Nr. 10/86 Oktober
DM 6.50, sfr 6.50, öS 50, Lit 5900, hfl 7.50

peeker

Antriebselektronik der Diskettenlaufwerke

Change-Befehl für BASIC

Bildschirmtreiber für Double-Hires-Schrift

1-Megabyte-RAM-Karte

Apple GS und Atari ST im Vergleich

GEM-Desktop und DOS-Shell beim Atari ST

Jetzt mit regelmäßigem Atari-ST-Sonderteil

Hüthig
PUBLIKATION

Ein Exemplar der hier gezeigten Bücher

Sie haben die Wahl!

erhalten Sie als "Dankeschön" für einen neuen Abonnenten, den Sie uns vermitteln.

Als »**peeker**«-Leser wissen Sie, wie gut Ihnen dieses Magazin beim Umgang mit Ihrem Apple oder Kompatiblen hilft.

Denn: Wer einen Apple oder einen Kompatiblen hat, der soll auch seinen »**peeker**« haben.

peeker 10/86

Bestellcoupon

Ich habe den neuen Abonnenten geworben und erhalte kostenlos eines der folgenden Bücher (bitte ankreuzen)

- ☐ Bühler, **Applesoft Basic**
- ☐ Meinhold, **Was ist Elektronik**
- ☐ Schilling, **EDV–kein Geheimnis**
- ☐ Stiehl, **Apple ProDOS Bd. 2**
- ☐ Meinhold, **Was ist Mikroelektronik**

Name, Vorname

Straße, Postfach

PLZ, Ort

X

Datum, Unterschrift

Ich bin der neue Abonnent. Bitte liefern Sie mir bis auf Widerruf, zumindest aber für 1 Jahr, »peeker« zum Jahresbezugspreis von DM 72,– (Ausland plus DM 18,– Porto) an folgende Anschrift:

Name, Vorname

Straße, Postfach

PLZ, Ort

X

Datum, Unterschrift

Gewünschte Zahlungsweise
☐ gegen Rechnung
☐ bargeldlos durch Bankeinzug

Konto-Nr. | Bankleitzahl

Geldinstitut

Vertrauensgarantie:
Diese Bestellung kann ich innerhalb einer Woche bei Dr. Alfred Hüthig Verlag GmbH, Im Weiher 10, 6900 Heidelberg 1 widerrufen. Zur Wahrung der Frist genügt die rechtzeitige Absendung. Ich bestätige die Kenntnisnahme mit meiner Unterschrift:

X

2. Unterschrift

Coupon ausschneiden oder kopieren und einsenden an:

»peeker«
Abonnementservice
Im Weiher 10
6900 Heidelberg 1

Hüthig PUBLIKATION

editorial

Ab der vorliegenden Peeker-Ausgabe berichten wir nicht mehr ausschließlich über die Apple-II-Serie, sondern auch zusätzlich über die Atari-ST-Reihe. Zur Erinnerung: In den Juli- und August-Heften hatten wir eine Umfrage gestartet, die Ihnen die Möglichkeit bot, auf die redaktionelle Ausrichtung des Peekers Einfluß zu nehmen. Dabei zeigten die Umfrage-Ergebnisse einen eindeutigen Trend in Richtung Atari ST (über 20 %). Sollten Sie sich nicht an der Umfrage beteiligt haben, so können Sie dies nachholen, indem Sie uns entweder eine Umfrage-Karte (aus dem Juli- oder August-Heft) oder eine Info-Karte (aus den nachfolgenden Heften) zusenden. Wir werden dann zum Jahresende eine aktualisierte Auswertung vornehmen.

Die 8-Bit-Ära, die zehn Jahre lang durch den Apple II maßgeblich bestimmt worden ist, wird mit dem Apple-II-Nachfolger, der den Apple IIe wohl nach einiger Zeit ablösen wird, ihren natürlichen Abschluß finden. Dies ist bedauerlich, da Schüler, Studenten und Hobbyisten nur bei 8-Bit-Rechnern die Chance hatten, ihre Maschine wirklich zu verstehen. Um beispielsweise das 8-Bit-ProDOS vollständig zu dokumentieren, mußte ein gerade noch überschaubares, 500seitiges Buch geschrieben werden (Schäpers: „ProDOS-Analyse"). Für eine ebenso akribische Analyse des 192K-Atari-ROMs wäre ein 6000seitiges Buch erforderlich. Wer würde das schreiben? Wer würde das lesen?

In diesem Heft bringen wir einen Vergleichsbericht zwischen dem Apple GS und dem Atari ST, der Ihnen die Kaufentscheidung erleichtern soll. Während der Atari ST in Deutschland bereits bestens eingeführt ist, muß der Apple GS erst noch seine Bewährungsprobe am Markt bestehen. Hierbei werden technische Details, etwa ob der Apple GS jetzt mit 2,8 oder später mit 4 MHz getaktet sein wird, kaum eine Rolle spielen. Entscheidend wird das „Marketing-Mix" sein, also die Mischung aus Preis-, Produkt, Werbe- und Vertriebspolitik. Ohne massives Marketing wird keine Marktakzeptanz zu erzielen sein. Ein untrügliches Zeichen für die Vitalität eines Mikrocomputers ist das Engagement deutscher Firmen für Zusatz-Hardware und -Software. Für den Atari ST gibt es beispielsweise sowohl ein deutsches BASIC als auch ein deutsches Pascal, d. h. Programmiersprachen, die von deutschen Firmen von Grund auf entwickelt worden sind. Beim Apple II tut sich in dieser Hinsicht seit zwei Jahren nichts mehr. Man hat sogar inzwischen Schwierigkeiten, Importeure für Programme amerikanischer Provenienz zu finden, von deutschen Produkten ganz zu schweigen. Dies ist auch ein Grund dafür, daß eine exklusive Behandlung des Apple II im Peeker nicht mehr möglich ist, denn es hat sich als widersinnig erwiesen, amerikanische Produkte vorzustellen, die man nirgendwo in Deutschland erwerben kann.

Die Zukunft des Apple GS wird entscheidend vom Marketing der Firma Apple sowie vom Engagement deutscher Soft- und Hardware-Häuser abhängen. Die Zeiten, als sich Mikrocomputer wie etwa der Apple II noch „von allein" verkaufen ließen, sind endgültig vorbei. In diesem Monat konnte man wieder Atari-Werbespots im ZDF sehen. Auch die Zeitschriftenanzeigen für den Atari ST sind kaum übersehbar. Wir wollen deshalb hoffen, daß die Firma Apple ähnliche Signale setzt.

Ulrich Stiehl

INHALT

Impressum

Peeker
3. Jahrgang 1986
ISSN 0176-9200
© für den gesamten Inhalt einschließlich der Programme Dr. Alfred Hüthig Verlag, Heidelberg 1986
Verleger und Herausgeber:
Dipl.-Kfm. Holger Hüthig
Geschäftsführung Zeitschriften:
Heinz Melcher
Chefredakteur: Ulrich Stiehl (us)
Redaktion: Dagmar Berberich
Anzeigenleitung: Karl M. Dietzow
Anzeigendisposition: Diana Walter

Telefonnummern:
Zentrale: 06221/489-0
Redaktion: 06221/489-352
Anzeigen: 06221/489-206
Abonnement: 06221/489-283
Software: 06221/489-231
Bücher: 06221/489-353
(Bestellungen bitte nur schriftlich)

Abonnement:
Der Abonnent kann seine Bestellung innerhalb von 7 Tagen schriftlich durch Mitteilung an den Dr. Alfred Hüthig Verlag GmbH, Postfach 102869, 6900 Heidelberg 1, widerrufen. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs (Datum des Poststempels). Das Abonnement verlängert sich zu den jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Jahresende schriftlich gekündigt wird. Die Abonnementsgelder werden jährlich im voraus in Rechnung gestellt, wobei bei Teilnahme am Lastschriftabbuchungsverfahren über die Postscheckämter und Bankinstitute eine vierteljährliche Abbuchung möglich ist.
Nichterscheinen infolge höherer Gewalt berechtigt nicht zu Ansprüchen gegen den Verlag.

Hüthig
PUBLIKATION

peeker

Heft 10/1986

Hardware

Aufbau und Funktion des Apple-Disk-II-Laufwerks
Teil 1: Mechanik und Antriebselektronik
von Dipl.-Ing. Gerhard Berg 6

Utility

Der Change-Befehl
Eine Utility für Applesoft-Programmierer
von Frank Seide 22

Applesoft

Hex-Monitor
Hilfsprogramm für die Ein- und Ausgabe von Hex-Dumps
von Günter Radestock und Ferdinand Stühlen 36

Grafik

Textzeichen auf dem Double-Hires-Bildschirm
von Steffen Holzinger 38

UCSD

Utility-Programme für UCSD-Pascal
Disk-Editor, Libary-Programm und Disassembler
von Dieter Geiß 44

Telekommunikation

Hallo, hier spricht der Apple
Erfahrungen mit einem Akustikkoppler
von Franz-Josef Hüskens 48

Bücher 51

Produkte

Seikosha-Drucker im Kurztest
von Andreas Schweitzer 55

CIRTECH-Flipper-Karte
1 MByte RAM für den Apple II+ und IIe
getestet von Hans-Martin Eng 56

Kyan-Pascal Programing Toolkit Advanced Graphics
getestet von Matthias Meyer 58

Firmenmitteilungen 60

Leserbriefe und Ergänzungen 62

Errata 64

Atari

Apple GS und Atari ST im Vergleich
von Ulrich Stiehl 65

Atari-Benutzerschnittstellen für Anwender und Programmierer
GEM-Desktop und DOS-Shell
von Ulrich Stiehl 71

Atari-Händler 77

Inserentenverzeichnis 78

Anschrift:
Dr. Alfred Hüthig Verlag GmbH
Im Weiher 10, Postfach 10 28 69
6900 Heidelberg
Telefon (0 62 21) 4 89-0
Telex 4-6 17 27 hued d.
Telefax (0 62 21) 489 279
BTX * 51851 #

Auslieferung für die Schweiz:
Delta-Verlag
Herr R. de Forest
Gugelmattstraße 31
8967 Widen
Telefon 057 / 33 86 86

Vertrieb:
Erscheinungsweise: 12 Hefte jährlich,
Erscheinungstag jeweils 1 Woche vor Monatsbeginn.
Jahresabonnement Inland DM 72,–, einschl. MwSt und Versandkosten.
Jahresabonnement Ausland DM 72,– plus DM 18,– Versandkosten.
Einzelheft DM 6,50
Vertrieb Handel:
MZV – Moderner Zeitschriften Vertrieb GmbH
Breslauer Str. 5, Postfach 1123,
8057 Eching b. München,
Tel. 089 / 31 90 06 13, Telex 0 522 656
Vertriebsleitung:
Walter Menzel, Tel. (0 62 21) 48 92 80

Bankverbindungen:
Zahlungen: an den Dr. Alfred Hüthig Verlag GmbH, D-6900 Heidelberg 1: Postgirokonten: Ludwigshafen 4799-673, BLZ 545 100 67; Österreich: Wien 75558 88; Schweiz: Basel 40-24417-4; Niederlande: Den Haag 1 457 28; Italien: Mailand 5 968 92 08; Belgien: Brüssel 07 230 26-85; Dänemark: Kopenhagen 603 4969; Norwegen: Oslo 199 4243; Schweden: Stockholm 5477 76-5
Bankkonten: Landeszentralbank Heidelberg 67 207 341; BLZ 672 000 00; Deutsche Bank Heidelberg 02 65 041; BLZ 672 700 03; Bezirkssparkasse Heidelberg 204 51, BLZ 672 500 20.

Herstellung:
Produktionsleitung: Gunter Sokollek
Gestaltung: Rainer Schmitt
Titelbild: Werner Hable
Satz und Druck:
Heidelberger Verlagsanstalt
Printed in Germany

Aufbau und Funktion des Apple-Disk-II-Laufwerks

Teil 1: Mechanik und Antriebselektronik

von Dipl.-Ing. Gerhard Berg

Im Rahmen der Artikelserie über Diskettensysteme beginnt in diesem Heft eine Beschreibung des Apple DISK II Disketten-Laufwerks. Während im Einführungsartikel (s. Peeker 3/85, S. 10-21), dessen Inhalt als bekannt vorausgesetzt wird, die Funktion von Disketten-Laufwerk und -Controller in allgemeiner Form erläutert wurde, wird in diesem Artikel das DISK II Laufwerk im Detail beschrieben. Im ersten Teil des Artikels wird die Mechanik und die Elektronik zur Ansteuerung der Motoren erläutert. Im zweitem Teil folgt dann die Beschreibung der Schreib-/Leseelektronik. Leider war bei der Vorbereitung des Artikels keinerlei Unterstützung von der Firma Apple zu erhalten. Deshalb basiert der Artikel größtenteils auf Untersuchungen des Autors und kann deswegen keinen Anspruch auf absolute Richtigkeit erheben.

Der Artikel verfolgt zwei Ziele. Zum einen will er Kenntnisse über Aufbau und Funktion des Diskettenlaufwerks vermitteln. Zum anderen will er Unterstützung bei Fehlersuche und Reparatur von defekten Laufwerken geben. In diesem Zusammenhang muß davor gewarnt werden, daß die in diesem Artikel beschriebenen Arbeiten am Laufwerk (Öffnen des Laufwerks, Änderung von Justagen etc.) eine Verletzung der Garantie darstellen können und deshalb nur an Laufwerken vorgenommen werden sollten, bei denen die Garantie schon abgelaufen ist. Ebenso müssen Autor und Verlag jegliche Verantwortung ablehnen, wenn nach Eingriffen in ein Laufwerk dieses evtl. nicht mehr einwandfrei oder gar nicht mehr funktioniert. Alle Arbeiten sollten nur von Personen mit ausreichender Erfahrung im Hardwarebereich vorgenommen werden.

1. Die Mechanik

1.1. Aufbau der Mechanik

Der mechanische Aufbau des DISK II Disketten-Laufwerks ist in den Bildern 1 bis 4 zu sehen. Zur Erläuterung sind in den Bildern die wichtigsten mechanischen und elektronischen Komponenten mit den im Text verwendeten Bezeichnungen versehen.

Bild 1 zeigt die Oberseite des Laufwerks, nachdem es aus dem Gehäuse ausgebaut wurde. Die große Leiterkarte, die auf das Laufwerk montiert ist, ist die sog. **Analogkarte**. Sie enthält die elektronischen Komponenten für alle Funktionen mit Ausnahme der **Motorregelung**. Die Komponenten für diese Funktion sind auf der in **Bild 2** gezeigten kleinen Leiterkarte auf der Rückseite des Gerätes enthalten.

Bild 3 zeigt die Oberseite des Laufwerks, nachdem die Analogkarte abgebaut wurde. **Bild 4** schließlich zeigt das Laufwerk von unten. Auf den beiden letzten Bildern kann man alle wichtigen mechanischen Komponenten sehen.

Zum besseren Verständnis des nachfolgenden Beitrags greife man auf die früheren Aufsätze „Aufbau und Funktion von Diskettensystemen" (Heft 3/85, S. 11 ff.) sowie „Testprogramme für Apple-II-Diskettenlaufwerke. Analyse von Disketten, Laufwerk und Controller" (Heft 8/85, S. 6 ff.) von G. Berg zurück.

Soweit wie Sie es auf den Bildern gesehen haben, kann man das Laufwerk ohne größeres Risiko zerlegen. Dabei muß unbedingt der Rechner ausgeschaltet werden, und man sollte sich den genauen Sitz aller Stecker merken, um beim späteren Zusammenbau die Stecker wieder richtig aufstecken zu können. Ein falsches Aufstecken eines Steckers kann zur Zerstörung von Komponenten im Laufwerk führen. Zum Öffnen des Laufwerks geht man folgendermaßen vor:

a. Die vier Schrauben an der Unterseite des Laufwerkes lösen.
b. Laufwerk nach vorne aus dem Gehäuse ziehen.
c. Schnappverschluß für das Flachbandkabel öffnen.
d. Die vier versenkten Schrauben an der Unterseite des Laufwerks lösen.
e. Untere Gehäuseplatte abnehmen.

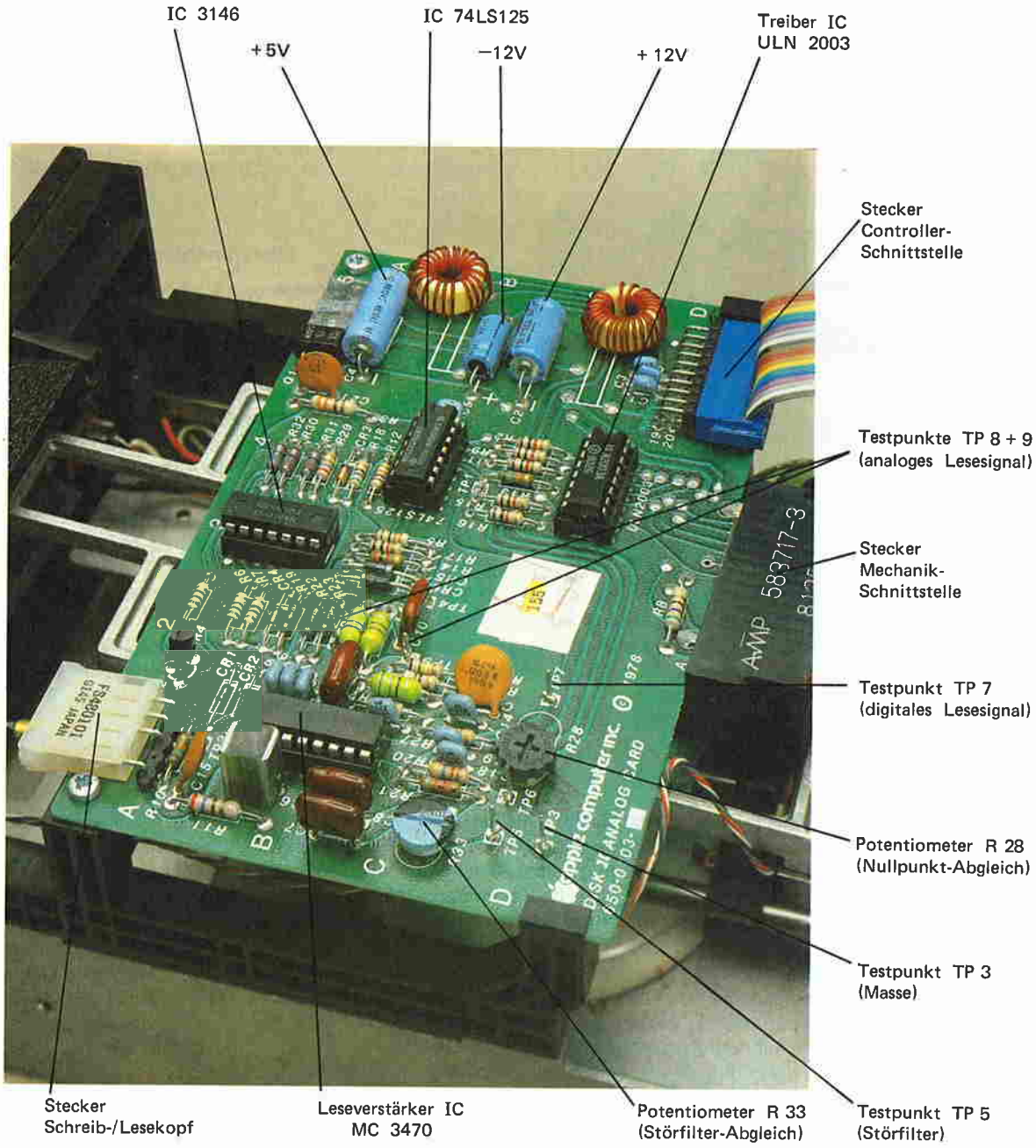


Bild 1: Laufwerk-Oberseite mit Analogkarte

f. Flachbandkabel-Stecker (Controller-Schnittstelle) abziehen. (Achtung: Dieser Stecker ist nicht codiert. Beim späteren Zusammenbau muß man genau darauf achten, daß er nicht versetzt oder verdreht aufgesteckt wird.)
g. Stecker für Schreib-/Lesekopf abziehen (Stecker mit vier dünnen verdrillten Leitungen).
h. Stecker am hinteren Rand der Analogkarte (Mechanik-Schnittstelle) abziehen. (Beim späteren Zusammenbau müssen die beiden Plastik-Klammern im Stecker in die Löcher in der Leiterkarte einrasten.)
i. Die zwei Schrauben lösen, die die Analogkarte halten.
j. Analogkarte nach vorne aus der Führung ziehen.

Nachdem das Laufwerk so demontiert wurde, kann man alle Komponenten des Laufwerkes direkt sehen. Für Messungen an der Analogkarte sind lediglich die Schritte a. und b. und für Messungen an der Motorregelung die Schritte a. bis e. erforderlich. Beim späteren Zusammenbau erfolgen alle Schritte genau in umgekehrter Reihenfolge.

1.2. Funktion der Mechanik

Wenn die Frontklappe des Laufwerks geöffnet ist, kann eine Diskette entlang der beiden Plastik-Führungsschienen in das Laufwerk geschoben werden. Beim Schließen der Frontklappe wird der mit zwei Blattfedern befestigte Andruck-Rahmen nach unten bewegt. Dadurch taucht der Spreizkonus durch die Einspannöffnung der Diskette in den Aufnahmekonus. Bei diesem Vorgang wird die Diskette zuerst zentriert und dann zwischen Spreizkonus und Aufnahmekonus eingespannt.

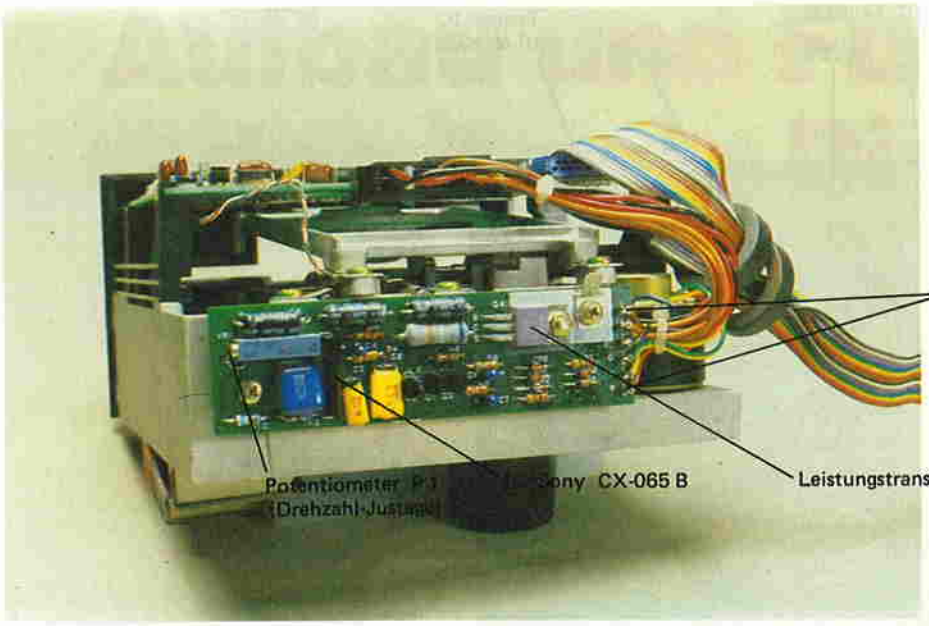


Bild 2: Laufwerk-Rückseite

Auf der gleichen Achse, die den Aufnahmekonus trägt, sitzt auf der Unterseite des Laufwerkes die große Riemenscheibe. Wenn der Motor läuft, wird über die beiden Riemenscheiben und den Antriebsriemen die Diskette in der Hülle gedreht. Die beiden Riemenscheiben sind leicht bauchig. Dadurch wird erreicht, daß der Riemen immer in der Mitte der Riemenscheiben läuft und nicht von ihnen fällt. Auf die große Riemenscheibe ist eine Stroboskopscheibe aufgeklebt, die zur groben Drehzahljustage benutzt werden kann (s. Abschnitt 4.1.).

Die Bewegung des Andruckrahmens hat noch eine zweite Funktion. Wenn die Frontklappe des Laufwerks geöffnet ist, dann ist der Kopfandruckhebel vom Schreib-/Lesekopf abgehoben. Dadurch ist genug Platz vorhanden, um die Diskette zwischen Kopf und Andruckfilz hindurchschieben zu können. Wenn die Frontklappe geschlossen wird, dann wird der Kopfandruckhebel freigegeben und mit Federspannung gegen den Schreib-/Lesekopf gedrückt. Um die Diskette nicht zu verkratzen, ist auf den Andruckhebel ein Filzplättchen geklebt, das die Diskette leicht gegen den Schreib-/Lesekopf drückt. Damit wird erreicht, daß die Diskette in ständigem Kontakt mit dem Schreib-/Lesekopf ist, was für eine einwandfreie Funktion des Schreibens und Lesens unbedingt erforderlich ist (s. Peeker 3/85). Der Luftspalt des Kopfes ist nur wenige Mikrometer breit und mit bloßem Auge nicht zu sehen. Er liegt etwa in der Mitte des schwarzen Striches, der auf dem Kopf sichtbar ist.

Der Schreib-/Lesekopf muß beweglich sein, damit man ihn auf die verschiedenen Spuren der Diskette positionieren kann. Er ist auf dem Kopfschlitten montiert, der auf zwei Führungsstangen hin und her bewegt werden kann. Die Bewegung des Kopfschlittens erfolgt über einen Mitnehmer, der ähnlich einer Grammophonnadel in der spiralförmigen Rille der Spiralscheibe läuft. Durch diese Anordnung wird die Drehbewegung des Schrittmotors in eine lineare Bewegung des Kopfschlittens umgewandelt. Wenn das Laufwerk nicht eingeschaltet ist, kann durch vorsichtiges Drehen der Spiralscheibe der Kopf hin und her bewegt werden. Sollte einmal der Mitnehmer aus der Rille gesprungen sein, braucht die Spiralscheibe nur solange gedreht zu werden, bis der Mitnehmer wieder einrastet.

Auf den Bildern nicht sichtbar, befindet sich auf der linken Seite des Laufwerks schließlich noch der Mikroschalter, der feststellt, ob die eingelegte Diskette schreibgeschützt ist oder nicht. Ebenso auf den Bildern nicht zu sehen, sitzt in der Frontplatte die Betriebsanzeige-LED, die aufleuchtet, wenn das Laufwerk vom Rechner angesprochen (selektiert) wird.
Im Gegensatz zu zahlreichen anderen Laufwerken verfügt das Apple DISK II Laufwerk weder über einen Mechanismus zum Erkennen der Spur Null (Schalter oder Lichtschranke) noch über eine Lichtschranke zum Erkennen des Indexloches in der Diskette.

2. Die Schnittstellen

Das DISK II Laufwerk hat vier Schnittstellen, eine externe und drei interne. Die externe Schnittstelle enthält alle Leitungen, die nach außen zum Controller führen. Die internen Schnittstellen dienen zur Verbindung der Laufwerkskomponenten untereinander. Der Stecker, der am hinteren Rand der Analogkarte in der Mitte sitzt, stellt die Verbindung zu den verschiedenen elektromechanischen Komponenten des Laufwerks (Schalter, Motoren usw.) her. Diese Schnittstelle wird deshalb kurz Mechanik-Schnittstelle genannt. Die beiden anderen internen Schnittstellen enthalten die Verbindungen zur Motorregelung und zum Schreib-/Lesekopf.

Die Kontaktbelegung der Schnittstellen-Stecker ist in Tabelle 1 bis 4 enthalten. In den Tabellen sowie im späteren Text und den Schaltbildern wurden (soweit vorhan-

den) die gleichen Signalnamen wie im DOS-Handbuch verwendet. Da die Signalnamen Abkürzungen der englischen Funktionsbezeichnung sind, ist zum leichteren Verständnis die Funktionsbeschreibung in den Tabellen in englischer und deutscher Sprache enthalten.

2.1. Controller-Schnittstelle

Das Apple DISK II Laufwerk wird über ein 20-poliges Kabel an die Interfacekarte (Controller) angeschlossen. Über das Kabel werden sowohl die Stromversorgungs- als auch die Signalleitungen geführt. Zur Entstörung wird das Flachkabel durch zwei Ferritringe geführt. Die Masseleitung (GND = Ground) wird für Stromversorgung und Signalleitungen gemeinsam benutzt. **Tabelle 1** zeigt die Stiftbelegung für den Stecker der Controller-Schnittstelle. Der Stecker enthält in der oberen Reihe die ungeraden und in der unteren Reihe die geraden Stifte. Für die Stifte 1, 2, 19 und 20 sind die Positionen auf der Leiterkarte aufgedruckt.

Zur **Stromversorgung** benötigen die Apple DISK II Laufwerke die Spannungen +5 V, +12 V und -12 V. Wenn das Laufwerk nicht benutzt wird (Standby), beträgt die Stromaufnahme bei +5 V ca. 130 mA , bei +12 V ca. 50 mA und bei -12 V ca. 3,5 mA. Während des Betriebs (Schreiben und Lesen) erhöht sich die Stromaufnahme der +12 V-Spannung auf ca. 550 mA.

Zur Steuerung des Apple DISK II Laufwerks werden 9 Signalleitungen benötigt. Alle Signale werden von einfachen LS-TTL-Bausteinen und nicht von besonderen Leitungstreibern angesteuert. Deshalb muß die Kabellänge zwischen Controller und Laufwerk auch recht kurz sein und darf nicht verlängert werden.
Mit dem Signal **-ENBL** (Enable = Freigabe) wird das Laufwerk eingeschaltet (selektiert). Das ist daran zu sehen, daß die Betriebsanzeige-LED aufleuchtet und der Antriebsmotor läuft.
Die Signale Φ0, Φ1, Φ2 und Φ3 dienen unmittelbar zur Ansteuerung der vier Phasen des Schrittmotors (s. Peeker 3/85, S.13/16).

Mit dem Signal **-WR REQ** (Write Request = Schreib-Befehl) wird das Schreiben eingeschaltet (vorausgesetzt das Laufwerk ist selektiert und die eingelegte Diskette nicht schreibgeschützt). Wenn das Laufwerk nicht auf Schreiben geschaltet ist, befindet es sich automatisch im Lese-Modus.
Das Signal **WR DATA** (Write Data = Schreib-Daten) dient beim Schreiben zur Übertragung der Daten vom Controller zum Laufwerk.
Das Signal **W PROT** (Write Protect = Schreibschutz) meldet an den Controller, ob die im selektierten Laufwerk eingelegte Diskette schreibgeschützt ist oder nicht.
Das Signal **RD DATA** (Read Data = Lese-Daten) dient beim Lesen zur Übertragung der Daten von dem gerade selektierten Laufwerk zum Controller.

2.2. Mechanik-Schnittstelle

Tabelle 2 enthält die Stecker-Belegung der Mechanik-Schnittstelle. Diese enthält die Anschlüsse für die Betriebsanzeige, den Schreibschutzschalter, den Schrittmotor sowie die Verbindungen zur Motor-

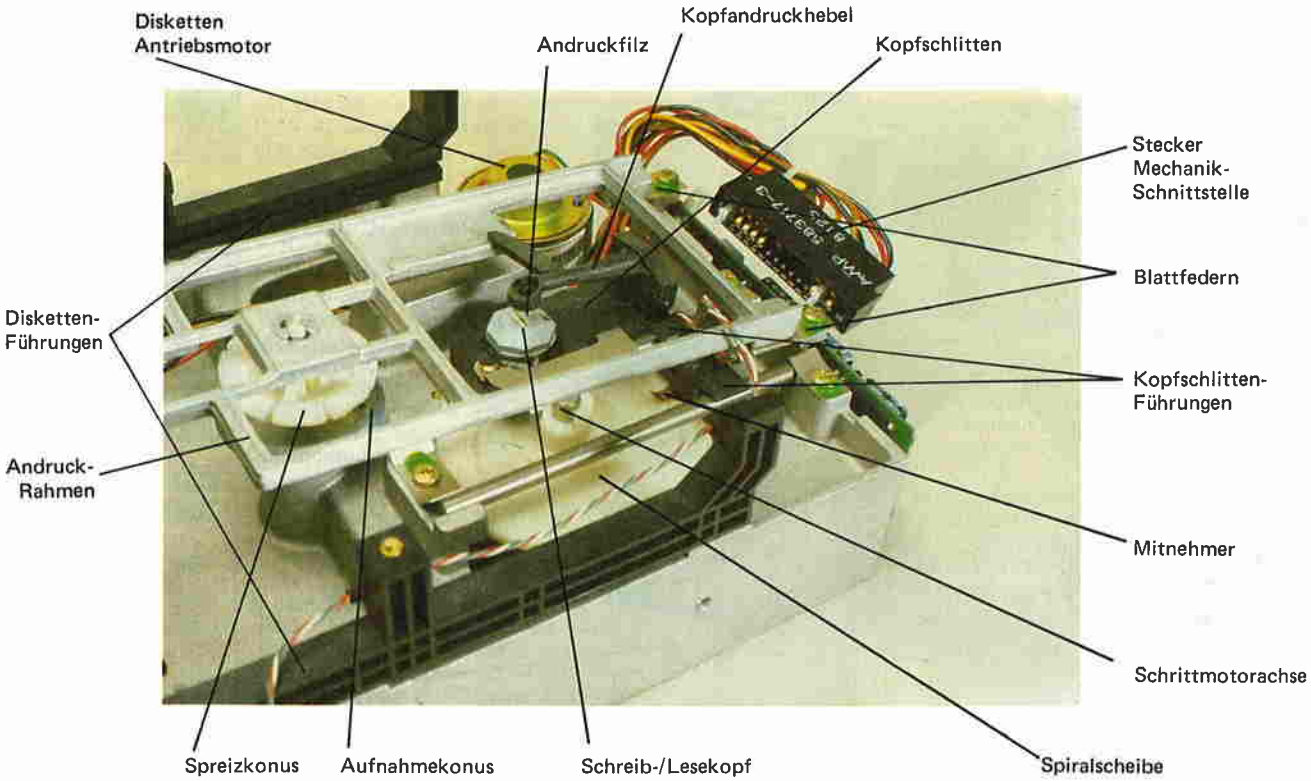


Bild 3: Laufwerk-Oberseite ohne Analogkarte

regelungs-Leiterkarte. Die Funktion der einzelnen Signale wird in der Beschreibung der jeweiligen Funktionsgruppen erläutert.
Beim Stecker der Mechanik-Schnittstelle sind die Kontakte auf der Oberseite der Leiterkarte mit Buchstaben und auf der Unterseite mit Zahlen gekennzeichnet. Die Position des Kontaktes A ist auf der Leiterkarte aufgedruckt. Der Kontakt 1 liegt genau unter dem Kontakt A.

2.3. Motorregelungs-Schnittstelle

Die Motorregelungs-Schnittstelle (**Tabelle 3**) enthält einerseits die Leitungen, die von der Mechanik-Schnittstelle der Analogkarte kommen und andererseits die Leitungen, die zum Antriebsmotor führen.
Die Anschlußleitungen werden nicht über einen Stecker, sondern über Lötstützpunkte geführt. Die auch auf der Leiterkarte aufgedruckten Stiftbezeichnungen sind Abkürzungen für die Kabelfarbe.

2.4. Schreib-/Lesekopf-Schnittstelle

Obwohl die Schreib-/Leseelektronik erst im zweiten Teil dieses Artikels beschrieben wird, ist der Vollständigkeit halber in **Tabelle 4** die Belegung des Schreib-/Lesekopf-Steckers angegeben. Die vier Anschlußleitungen des Schreib-/Lesekopfes werden über den kleinen Stecker am vorderen Rand der Analogkarte geführt. Stift 1 des Steckers liegt an der Außenseite der Leiterkarte. Position 2 des Steckers enthält eine Codierung, um den Stecker gegen falsches Aufstecken zu schützen.

3. Funktion der Elektronik

Die Elektronik des Apple DISK II Laufwerks besteht aus fünf Funktionsgruppen: Stromversorgung und Freigabe, Schrittmotor-Ansteuerung, Motorregelung, Schreibelektronik, Leseelektronik.
Die Motorregelung befindet sich auf der kleinen Leiterkarte an der Rückseite des Laufwerkes. Alle übrigen Funktionsgruppen befinden sich auf der größeren Leiterkarte auf der Oberseite des Laufwerkes. Die kleine Leiterkarte trägt beim Laufwerk des Autors die Bezeichnung PY117 und die große Karte die Bezeichnung „DISK II ANALOG CARD 650-0103". Das Schaltbild der Analogkarte ist dem Apple II DOS-Handbuch entnommen, jedoch zum besseren Verständnis umgezeichnet. In den Schaltbildern des DOS-Handbuchs haben einige Leitungen einen Strich über dem Signalnamen und einige nicht. Damit wird die Zuordnung zwischen logischen und physikalischen Zuständen definiert.

Auf logischer Seite gibt es die beiden Zustände 1 (aktiv) und 0 (inaktiv). Im aktiven Zustand wird die dem Signalnamen entsprechende Funktion (z.B. -MTR ON = Motor ein) ausgeführt. Im inaktiven Zustand wird die entsprechende Funktion nicht ausgeführt. Auf physikalischer Seite werden Spannungswerte gemessen. Bei den im Apple verwendeten TTL-Bausteinen werden Spannungen >= 2,0 V als H (High-Zustand) und Spannungen <= 0,8 V als L (Low-Zustand) definiert. Die logischen und physikalischen Zustände können einander auf zwei Arten zugeordnet werden.

Für Leitungen ohne einen Strich über dem Signalnamen gilt die sog. **positive Logik** (high-aktive Logik). Dabei entspricht der logische Zustand 1 dem physikalischen High-Zustand (H) und der logische Zustand 0 dem Low-Zustand (L), d.h. die dem Signalnamen entsprechende Funktion wird ausgeführt, wenn das Signal den H-Zustand (high) hat.

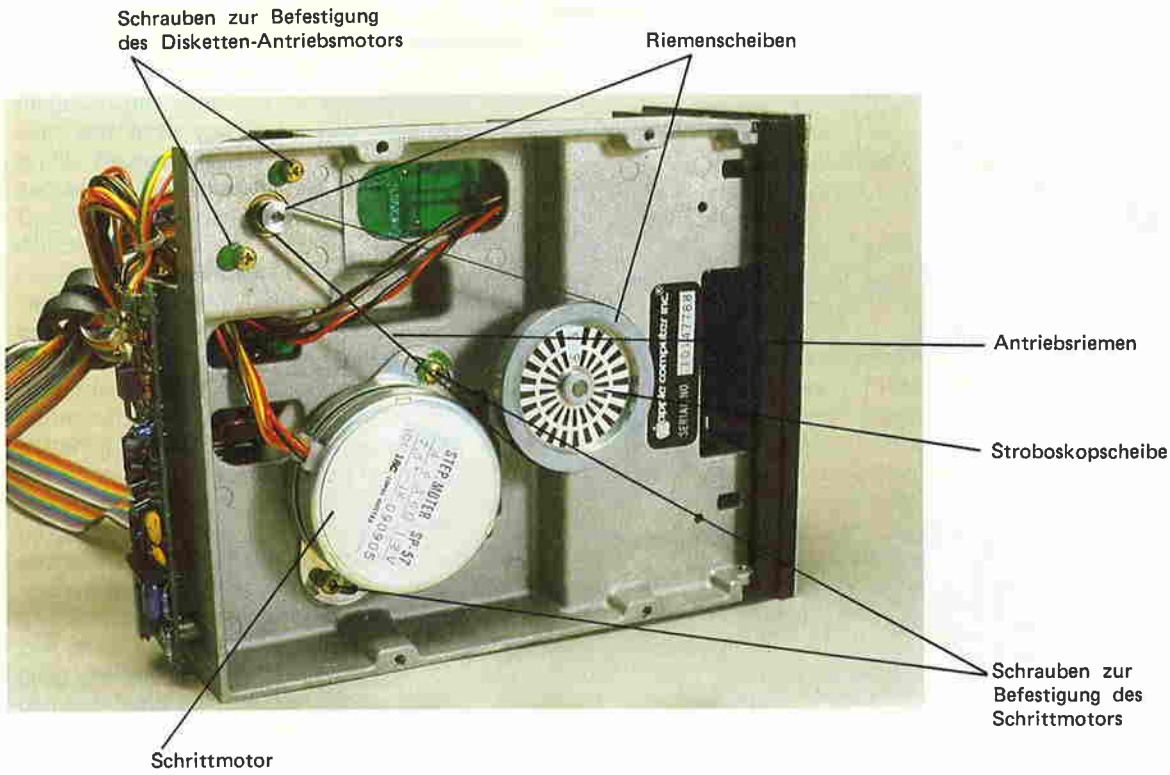


Bild 4: Laufwerk-Unterseite

MS BASIC ist eine höchst moderne und leicht erlernbare Programmiersprache!

210 reservierte Begriffe...Programmsynthese aus Modulen (lokale Variablen, Wertübergaben mit COMMON)...Nachladen von Segmenten (CHAIN, mit Parameterübergabe)...Einbinden von MAC-Funktionen (Graphik, Maus, Fenster, Knöpfe usw.)...Fremddateizugriff (von MS-BASIC auf Dateien von z.B. MULTIPLAN, MACPAINT, WORD)...Programmablaufsteuerung (Ereigniserfassung wie ON TIMER, ON MOUSE)...Peripheriebefehle (wie COMI für DFÜ)...strukturierte BASIC-Programmierung ohne Zeilennummern...Gleitkommaarithmetik.....

MACINTOSH und MS BASIC bilden eine komfortable Programmierumgebung!

Mehrfachfenster für Simultanbeobachtung (z.B. Fenster 1: Hauptprogramm, Fenster 2: Unterprogramm; oder Fenster 1: Listing, Fenster 2: Ergebnisse)...Mausedition wie in Textprogrammen...Collagetechnik (Programmabschnitte ausschneiden und versetzen)...

HIER DER KURSTEXT einer lebendigen und systematischen BASIC-Einführung von dem US-Professor David Lien. Ein Text für das Selbststudium, das in anspruchsvolle BASIC-Programmierungen mit den Funktionen des MACINTOSH mündet. Ideal als Schultext.
450 Seiten, Softcover, DM 59,–

tewi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

Weitere te-wi-Bücher

M68000 FAMILIE, 2 Bd.
Hilf/Nausch, ges. 968 Seiten
Einzige Motorola-authentische Darstellung von CPU-68000-Architektur, Programmierung, Systemaufbauten. Behandelt alle 68000-Bausteine sowie 68020, 68881.
Bd 1, Grundlagen + Architektur, 568 Seiten, DM 79,–
Bd 2, Anwendung und Bausteine, 400 Seiten, DM 69,–

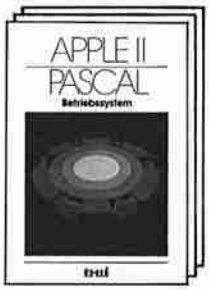
Das APPLE II/II+/IIe/IIc-Handbuch
L. Poole
Erst mit Hilfe dieses Leitfadens werden Sie Ihren Apple II erfolgreich einsetzen, denn Text und Bildmaterial gehen weit über das hinaus, was herstellerseitig an Literatur angeboten wird.
Neu überarbeitet und jetzt um die spezifischen Eigenheiten der Modelle **IIe** und **IIc** erweitert. 472 Seiten, Softcover, DM 66,–

APPLEWORKS integriert in APPLE II, IIe, IIc die Funktionen eines modernen Schreibtisches: Textverarbeitung, Datenbank, Rechenblatt, Datenfernübertragung. Sämtliche System/Anwendungsfragen in 2 Bänden.
Von Botta/Lange/Zimmermann, je 264 Seiten und je DM 49,–

tm 4601

DAS C-BUCH. **NEU**
Textbuch für C-Kurse und C-Anwendungen auf PCs. Beschreibt sämtliche Konstrukte der C-Sprache unter den Betriebssystemen MS DOS, CP/M, ISIS, UNIX und für die C-Compiler von MS, DR, LATTICE, INTEL. Didaktisch und typographisch außergewöhnlich. Mit über 100 lauffähigen Beispielprogrammen für PCs. Zeigt Realisierungen neuester Softwarestrategien in „C".
Von Herold/Unger. Herbst 86.
Etwa 500 Seiten. Softcover. DM 79,–

UMWELTDYNAMIK
30 Programme für kybernetische Umwelterfahrungen auf allen BASIC-Rechnern. Das Buch enthält beides: Ein Programmsystem zur Simulation eigener Problemformulierungen und 29 kommentierte Modellbeispiele wie Baumsterben, Heizungsbedarf, Nahrungsketten usw. Prospekt anfordern.
Von Hartmut Bossel, 480 Seiten, Softcover, DM 59,–



Erstes deutsches Referenzwerk
sämtlicher Befehle und Systemroutinen von Apple II, IIplus, IIe
APPLE II PASCAL
Betriebssystem, 272 S., DM 49,–
Sprache, 216 S., DM 39,–
Pascal 1.2 Addendum, 112 S., DM 36,–

Grundlagenbuch, Bestseller
APPLE II PASCAL.
Eine praktische Anleitung,
544 S., DM 59,–

Für Leitungen mit einem Strich über dem Signalnamen gilt die **negative Logik**. Da im laufenden Text die Überstreichung Schwierigkeiten bereitet, werden in diesem und den folgenden Artikeln Signale mit negativer Logik mit einem vorangestelltem Minuszeichen gekennzeichnet. Bei negativer Logik (low-aktive Logik) ist die Zuordnung genau umgekehrt zur positiven Logik, d.h. 1 entspricht L und 0 entspricht H. Die dem Signalnamen entsprechende Funktion wird also ausgeführt, wenn das Signal den L (low)-Zustand hat.

In den Schaltbildern werden Anschlüsse der Controller-Schnittstelle durch ein Rechteck gekennzeichnet. Durch Ovale werden Anschlüsse der internen Schnittstellen bezeichnet. Pfeile bezeichnen schließlich Verbindungen von einem Schaltbild zum anderen innerhalb der gleichen Leiterkarte.

3.1. Stromversorgung und Freigabe

Bild 5 zeigt die Stromversorgung und Freigabe-Schaltung des DISK II Laufwerks.
Die Versorgungsspannungen +5 V und +12 V werden über LC-Filter, die Spannung -12 V nur über Kondensatoren geglättet.
Die Freigabe (Selektion) des Laufwerks erfolgt über das Signal -ENBL (Enable). Wenn dieses (low-aktive) Signal im aktiven Zustand ist (1 = L), wird über die beiden invertierenden Treiber ULN 2003 der Leistungstransistor T1 und damit die Versorgungsspannung +12 V · ENBL eingeschaltet. Dadurch wird der Schrittmotor, die Schreibelektronik sowie die an der Frontplatte befindliche Betriebsanzeige-LED (Activity LED) eingeschaltet. Weiterhin geht das -ENBL-Signal über den Widerstand R9 als Signal -MTR ON zur Motorregelung, um den Disketten-Antriebsmotor einzuschalten.
Wenn das Signal -ENBL im inaktiven Zustand ist (0 = H), dann wird T1 und damit die Spannung + 12 V · ENBL ausgeschaltet. Dadurch werden alle Funktionen des Laufwerks wieder ausgeschaltet.

3.2. Schrittmotor-Ansteuerung

Bild 6 zeigt die Schrittmotor-Ansteuerung, die lediglich aus vier Treibern für die vier Spulen (Phasen) des Schrittmotors besteht. Als Treiber werden vier Elemente des integrierten Bausteins ULN 2003 verwendet. Dieser Baustein, der u.a. von den Firmen SGS, Signetics, Sprague und Texas Instruments hergestellt wird, enthält sieben gleiche Darlington-Treiber, wovon jeder Treiber eine maximale Spannung von 50 V und einen maximalen Strom von

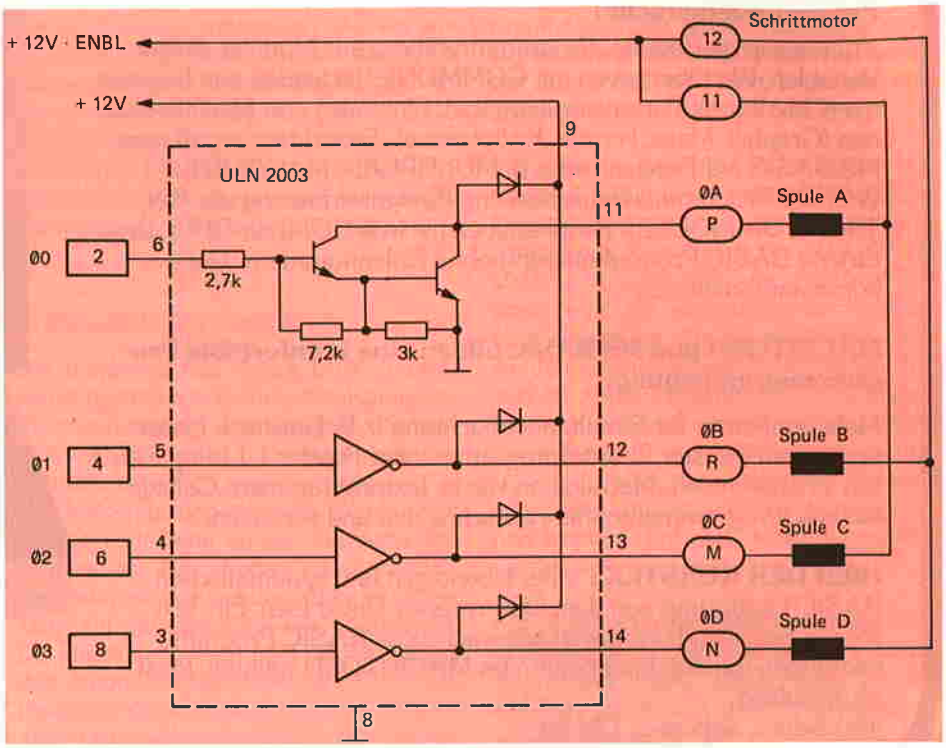


Bild 5: Stromversorgung (Hinweis: Ø = Phi)

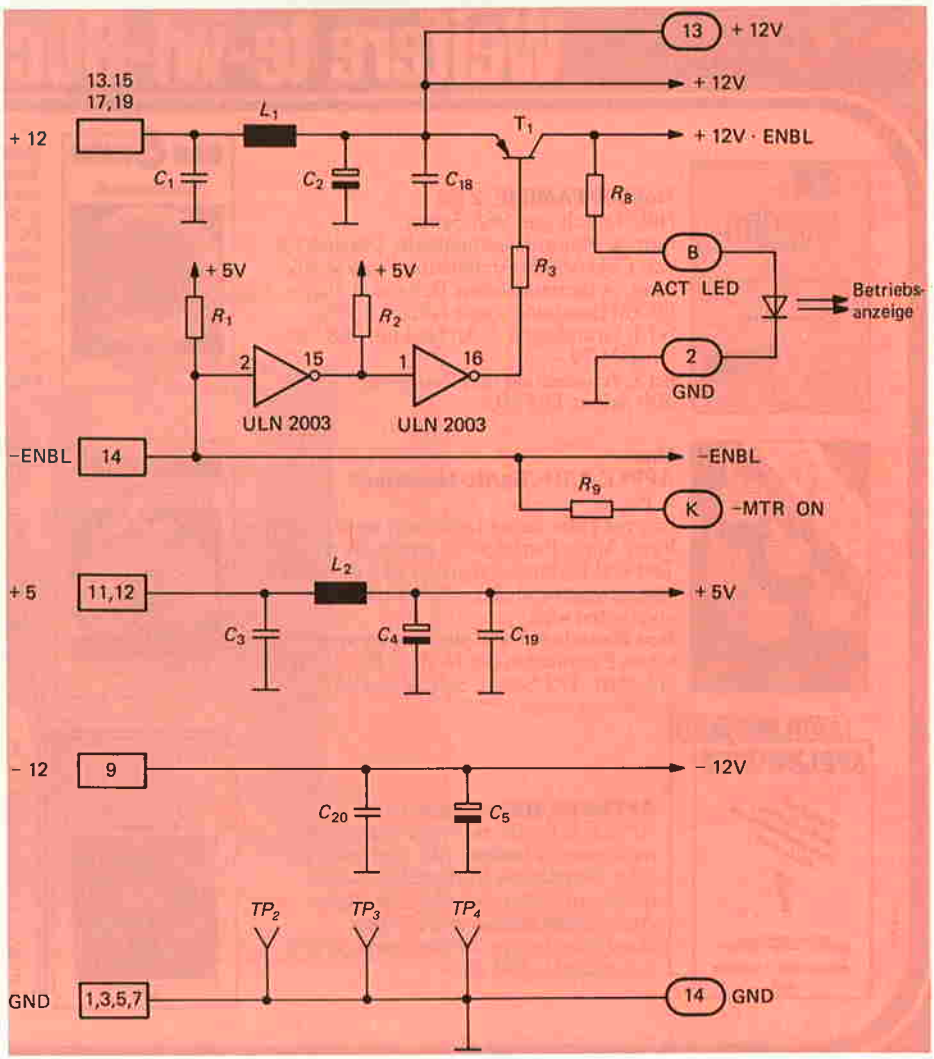


Bild 6: Schrittmotor-Ansteuerung

500 mA verträgt. Der Baustein enthält ebenfalls für jeden Treiber eine Freilaufdiode, deren Kathoden an den gemeinsamen Anschluß 9 gelegt sind. Das „Innenleben" des Bausteins ist am Beispiel von Φ0 in Bild 6 gezeigt.

Wenn ein Phasensignal (Φ0 bis Φ3) aktiv ist (1 = H), wird der zugehörige Darlington-Treiber eingeschaltet. Unter der Voraussetzung, daß mit dem Signal -ENBL die Spannung +12 V · ENBL eingeschaltet ist, fließt durch die entsprechende Spule des Schrittmotors (ΦA bis ΦD) ein Strom. Ist ein Phasensignal inaktiv (0 = L), so ist der entsprechende Treiber ausgeschaltet und durch die zugehörige Spule fließt kein Strom.

3.3. Motorregelung

Es gibt offensichtlich unterschiedliche Motorregelungs-Leiterkarten für das DISK II Laufwerk. So ist dem Autor bekannt, daß es abweichend von der hier beschriebenen Schaltung auch eine Schaltung gibt, die auf dem IC LM2917 basiert. Wann und in welchen Laufwerken welche Schaltung eingebaut wurde, ist dem Autor leider nicht bekannt, jedoch wäre er für entsprechende Informationen dankbar. Die Beschreibung kann sich deshalb nur auf die im Laufwerk des Autors enthaltene Schaltung beziehen. Das Herzstück dieser Schaltung ist der 8-polige integrierte Schaltkreis CX-065B der Firma Sony, der u.a. zur Antriebsregelung in Kasettenrekordern benutzt wird. **Bild 7** zeigt die im DOS-Handbuch nicht enthaltene Schaltung der Motorregelung.

Bild 8 zeigt den prinzipiellen Signalverlauf im Regelkreis. Im Schaltbild sind die Stellen, an denen die einzelnen Signale auftreten, mit den Buchstaben der jeweiligen Kurve gekennzeichnet.

Der in den Antriebsmotor integrierte Tacho-Generator erzeugt eine Sinusspannung (Kurve A), deren Frequenz proportional zur Drehzahl des Motors ist. Bei Nenndrehzahl beträgt die Periodendauer ca. 2,7 msec und die Spannung ca. 7,5 Vss (Volt von Spitze zu Spitze). Diese Sinusspannung wird über einen Vollweg-Gleichrichter (D1 bis D4) gleichgerichtet und an den Eingang 1 des Regel-ICs CX-065B gelegt. Der **Vollweg-Gleichrichter** wandelt die an seinen Eingängen liegende Wechselspannung (Kurve A) in eine pulsierende Gleichspannung doppelter Frequenz (Kurve B) um. Die Periodendauer der Kurve B beträgt ca. 1,35 msec und die Spannung ca. 2,5 Vss. Da dem Gleichrichter keine Kondensatoren zur Glättung nachgeschaltet sind, wird keine Gleichspannung erzeugt.

Innerhalb des Regel-ICs wandelt zunächst eine **Impulsformer**-Stufe die pulsierende Gleichspannung in Rechteck-Impulse um (Kurve C). Jedesmal, wenn die in Kurve B gestrichelt gezeichnete Schwelle überschritten wird, ist das Signal in Kurve C gleich 1. Die Rechteck-Impulse gelangen auf den Eingang des **Sägezahn-Generators**. Dieser wird mit jeder ansteigenden Flanke des Rechteck-Signals gestartet. Das Ausgangssignal des Sägezahn-Generators (Kurve D) gelangt auf den „+" Eingang eines **Komparators** und wird dort mit einer am „-" Eingang anliegenden Schwellenspannung verglichen. Wenn die Sägezahn-Spannung größer als die Schwellenspannung ist, wird der Ausgang des Komparators (Kurve E) zu 1. Während dieser positiven Impulse wird der Kondensator C4 über einen **Gleichspannungsverstärker** aufgeladen und in den übrigen Zeiten entladen (Kurve F). Über R4 und C8 wird diese Spannung geglättet. Die Spannungshöhe des geglätteten Signals entspricht dem Mittelwert des Signals in Kurve F und damit dem Einschaltverhältnis des Signals in Kurve E. Je länger das Signal in Kurve E eingeschaltet ist, desto höher ist die Spannung an C8. Von C8 aus wird schließlich über den Treiber-Transistor T2 der Leistungstransistor T4 angesteuert, der den Strom für den Antriebsmotor liefert. Dabei ist der Motorstrom umso größer, je höher die Spannung an C8 ist.

Um den Motor vor Überhitzung zu schützen, wird über R9 und T3 der Motorstrom begrenzt. Normalerweise ist T3 gesperrt. Wenn jedoch der Motorstrom zu hoch wird, wird der Spannungsabfall an R9 so groß, daß die erforderliche Basis-Emitter-Spannung erreicht und damit T3 leitend wird. Dadurch wird die Spannung an der

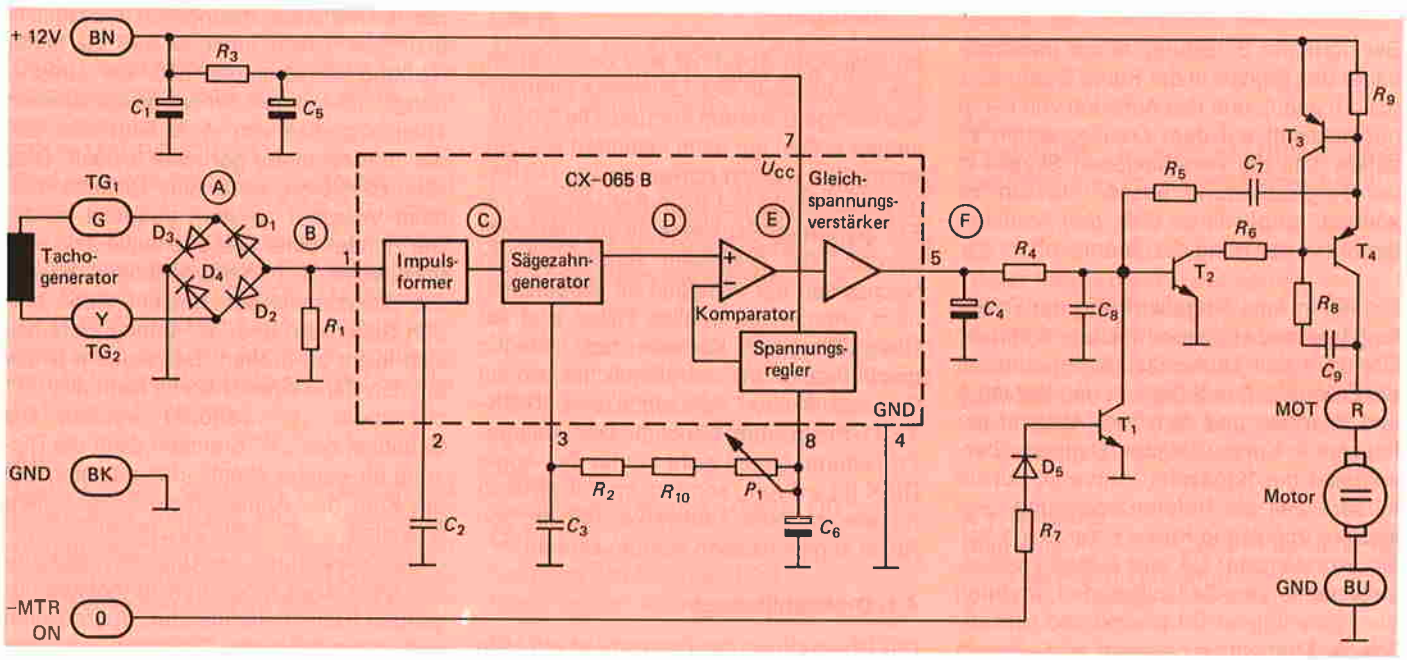


Bild 7: Motorregelung

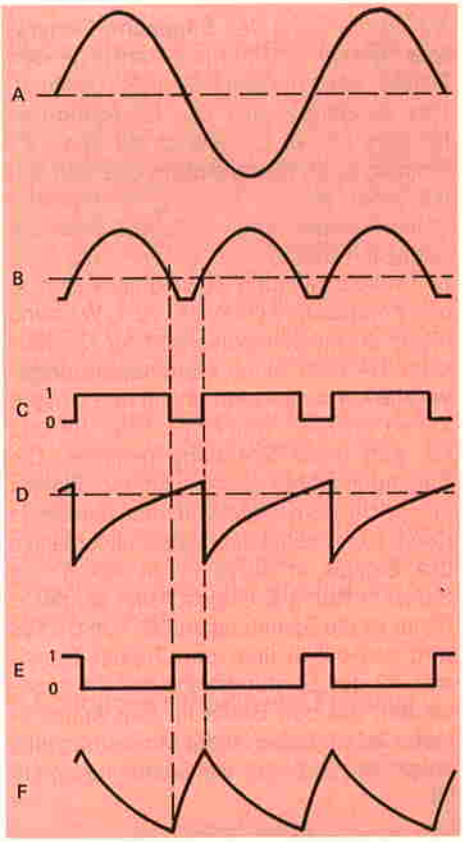


Bild 8: Signalverlauf im Regelkreis

Basis von T4 (gegenüber dem Emitter) reduziert und damit der Strom durch T4 und den Antriebsmotor begrenzt. Die beiden RC-Glieder R5-C7 und R8-C9 dienen der Stabilität der Regelschaltung.

Bei normaler Belastung ist die Einschaltdauer des Signals in der Kurve E sehr kurz (ca. 10 µsec), und das Aufladen von C4 ist nur schlecht auf dem Oszillographen zu sehen. Um die verschiedenen Signale in der Regelschaltung besser messen zu können, empfiehlt es sich, den Antriebsmotor mit der Hand abzubremsen.

Die eigentliche **Regelwirkung** der Schaltung geschieht folgendermaßen: Wenn die Drehzahl des Laufwerks größer wird, ist die Frequenz des Signals in den Kurven A und B größer und damit der Abstand der Impulse in Kurve C kürzer. Dadurch überschreitet der Sägezahn (Kurve D) nur für kürzere Zeit die Referenzspannung, was kürzere Impulse in Kurve E zur Folge hat. Der Kondensator C4 wird dadurch seinerseits nur kürzere Zeit aufgeladen, wodurch die Spannung an C8 absinkt und schließlich der Motorstrom niedriger wird, was ein Absinken der Drehzahl zur Folge hat.

Bei zu niedriger Drehzahl ist der Vorgang genau umgekehrt. Die niedrigere Frequenz des Eingangssignals hat längere Impulse in Kurve E und damit eine höhere Spannung an C8 zur Folge. Durch den daraus resultierenden höheren Motorstrom wird der Motor beschleunigt, bis er die Nenndrehzahl erreicht hat.

Die Nenndrehzahl des Motors kann mit Hilfe des Potentiometers P1 eingestellt werden. P1 bestimmt (zusammen mit R2 und R10) die Größe des Stromes, mit dem C3 aufgeladen wird, was der Steilheit des Anstiegs des Sägezahns entspricht. Wenn der Widerstandswert von P1 verkleinert wird, so wird C3 schneller aufgeladen, wodurch die Referenzspannung eher erreicht wird. Dadurch werden die Impulse in Kurve E länger, und die Spannung an C8 und der Motorstrom steigen, wodurch der Motor beschleunigt wird, bis er die jetzt höhere Nenndrehzahl erreicht hat. Umgekehrt kann durch Erhöhen des Widerstandswertes von P1 die Nenndrehzahl erniedrigt werden.

Das Ein- und Ausschalten des Antriebsmotors erfolgt mit Hilfe des Signals -MTR ON (Motor On = Motor ein). Im aktiven Zustand des Signals (1 = L) ist der Transistor T1 nichtleitend, und die Regelschaltung funktioniert wie oben beschrieben. Im inaktiven Zustand (0 = H) ist der Transistor T1 leitend, wodurch die Transistoren T2 und T4 ausgeschaltet werden. Damit fließt kein Strom mehr durch den Motor; der Motor bleibt stehen.

4. Justagen

Im folgenden Abschnitt wird beschrieben, wie die Justagen des Laufwerks überprüft und korrigiert werden können. Die Einstellungen sollten nur dann verändert werden, wenn dies wirklich notwendig ist. Normalerweise ist das nur nach Reparaturen der Fall, bei denen Bauelemente ausgetauscht wurden. Bei normalem Betrieb ist ein Nachstellen der Justagen im allgemeinen nicht erforderlich. In allen Fällen wird zur Überprüfung und Korrektur der Justagen das in Peeker 8/85 veröffentlichte und auf Sammeldiskette #8 enthaltene **DISKTEST-Programm** benötigt. Die Justage-Prozeduren gelten nicht nur für das Apple DISK II Laufwerk, sondern im Prinzip auch für alle anderen Laufwerke, die an den Apple angeschlossen werden können.

4.1. Drehzahljustage

Die Überprüfung der Drehzahl ist mit Hilfe des DISKTEST-Programms ganz einfach. Der U-Befehl zeigt direkt die Abweichung von der nominalen Umdrehungsdauer grafisch und numerisch an. Zur Korrektur der Drehzahl wird das Potentiometer P1 auf der Motorregelungs-Leiterkarte (s. Bild 2) soweit verdreht, bis die gewünschte Drehzahl erreicht ist.

Bei der Drehzahljustage ist eine Reihe von Punkten zu beachten:

– Bei der Messung der Drehzahl mit dem DISKTEST-Programm wird die benutzte Diskette überschrieben!

– Die Drehzahlabweichung kann mit der Betriebstemperatur des Laufwerks schwanken. Diese Schwankungen können für unterschiedliche Laufwerkstypen mehr oder weniger groß sein. (Beim Apple DISK II Laufwerk sind sie recht klein.) Die Messung der Drehzahl sollte bei kaltem Laufwerk beginnen und über einige Zeit durchgeführt werden, bis sich das Laufwerk erwärmt hat und sich die Drehzahl nicht mehr ändert.

– Die Drehzahl kann je nach der Spur, auf welcher der Kopf gerade steht, unterschiedlich abweichen. Kopf und Andruckfilz bremsen durch ihre Reibung die Diskette ab. Je nach Spurposition ergibt sich dafür ein unterschiedlich langer Hebelarm, was ein unterschiedliches Bremsmoment zur Folge hat. Je nach Regelschaltung hat dies eine mehr oder weniger starke Drehzahlabweichung zur Folge. Um einen Mittelwert zu erhalten, wird deshalb empfohlen, die Drehzahljustage auf Spur 17 durchzuführen.

– Ebenso ist das Bremsmoment von Diskette zu Diskette unterschiedlich. Leichtgängige Disketten resultieren in einer etwas höheren und schwergängige Disketten in einer etwas niedrigeren Drehzahl. In extremen Fällen kann es bei zu hoher Reibung vorkommen, daß die Umdrehungsdauer unter der vorgeschriebenen Toleranzgrenze von -1 % liegt oder daß der Antriebsmotor gar nicht losläuft. Dies kann passieren, wenn eine Diskette (z.B. beim Versand) zu stark gedrückt wurde. Die Ränder einer Diskettenhülle sind normalerweise U-förmig umgebogen, so daß die Diskettenscheibe zwischen den beiden Schenkeln des „U" genug Platz hat, sich leicht zu drehen. Bei starkem Druck auf den Rand einer Diskette kann das „U" zu einem „V" verformt werden. Die Schenkel des „V" bremsen dann die Diskette ab – unter Umständen so stark, daß die Kraft des Antriebsmotors nicht mehr ausreicht, die Diskette zu drehen.

– Die Abweichung der Umdrehungsdauer soll unter allen möglichen Betriebsbedingungen (Betriebstemperatur, Spurposition und unterschiedliche Disketten) zwischen +1 % und −1 % liegen. Um sicherzustel-

Macroeditor

Durch den im Heft 10/1986, S. 22 ff. beschriebenen Change-Befehl wird der Applesoft-Macroeditor um einen weiteren, sehr mächtigen Befehl erweitert. Man starte unter DOS 3.3 zunächst den Macroeditor von Sammeldisk #16 mit BRUN MACROEDITOR und erst dann den Change-Befehl von Sammeldisk #22 mit BRUN CHANGE.REMS.DOS.3.3. Damit wird der Change-Befehl in den Macroeditor integriert. Mit CALL −151, 76FD: 4C 00 7D, Ctrl–C, BSAVE EDITOR.NEU, A$76FD, L$1F03 wird der erweiterte Macroeditor abgespeichert.
Es wurde übrigens früher vergessen darauf hinzuweisen, daß der Macroeditor auch über einen Merge-Befehl verfügt, womit sich Haupt- und Unterprogramm vereinen lassen: Zunächst Hauptprogramm laden, dann **&P,** dann Unterprogramm laden, dann **&M.**
Bestellhinweis: Der Macroeditor ist eine mit der Firma Heyden Datasystems vereinbarte kostenlose Zugabe, die sich nicht auf der normalen Disk #16 für Einzelbezieher befindet. Die früheren Fortsetzungsbezieher haben den Macroeditor automatisch kostenlos mit Disk #16 im April 1986 erhalten. Neue Fortsetzungsbezieher können Disk #16 (Vermerk: Mit Macroeditor!) rückwirkend für DM 20,– bestellen. Ferner kann man als Nicht-Fortsetzungsbezieher für DM 120,– sechs einzelne Sammeldisketten *gemischt* bestellen, z. B. #16, #22 sowie 4 weitere Disketten nach eigener Wahl.

Kurzanleitung

Der Applesoft-Editor setzt einen Apple II+/e/c (40 Z/Z) mit einem Laufwerk voraus und ist für 48K-DOS-3.3 (oder Diversi-DOS usw.), nicht jedoch für ProDOS gedacht. (Hierfür gibt es den analogen PRODOS.EDITOR über den Hüthig Software Service.) Nach dem Booten von DOS 3.3 oder Diversi-DOS legen Sie die Sammeldiskette #16 ein und starten Sie den Editor mit
BRUN MACROEDITOR
Eine spezielle Schnell-Lade-Routine stellt selbst unter dem alten DOS 3.3 sicher, daß der Editor in maximal 2-3 Sekunden im Speicher installiert ist. Der Editor belegt den Bereich $7D00-$95FF ($7D00 = dezimal 32000) und kann auch gestartet werden, wenn sich ein Applesoft-Programm bereits im Speicher befindet. Und nun die Befehle im einzelnen:

&U
Damit läßt sich der Editor vorübergehend deaktivieren, um z.B. ein Applesoft-Programm mit RUN auszutesten, das jedoch nicht HIMEM höher als 32000 oder MAXFILES auf einen anderen Wert als 3 setzen sollte, weil sonst der Editor überschrieben werden würde. Das gleiche gilt für FP; dagegen ist NEW natürlich erlaubt.
&
Damit oder auch mit CALL 32000 wird der Editor wieder aktiviert. Nunmehr kann man alle nachfolgenden Befehle eingeben.

Z
Es erscheint das Wort „LINE", und man muß jetzt die Nummer der zu ändernden Programmzeile eingeben, die dann auf den letzten 6 Zeilen des Bildschirms angezeigt wird. Als spezielle Korrigierbefehle sind implementiert:
Ctrl-B
Sprung zum Beginn der Zeile.
Ctrl-E
Sprung zum Ende der Zeile.
Ctrl-Z
Eine Zeile tiefer.
Ctrl-T
Tabulieren, d.h. 10 Zeichen vorwärts.
Linkspfeil
Ein Zeichen nach links.
Rechtspfeil
Ein Zeichen nach rechts.
Ctrl-D
Delete bzw. Löschen des Zeichens, auf dem der Cursor gerade ruht.
Ctrl-I
Insert bzw. Einschalten des Einfüge-Modus ab der momentanen Cursor-Position. Jedes nunmehr eingegebene Zeichen schiebt die Applesoft-Zeile entsprechend auseinander. Außerdem können im Einfüge-Modus die meisten Ctrl-Zeichen eingegeben werden außer
Rechtspfeil, Linkspfeil, Return,
mit denen man den Einfüge-Modus wieder aufhebt.
Return
Übernahme der Applesoft-Zeile, so wie man sie am Bildschirm sieht, in den Programmspeicher.
Ctrl-P
Abhacken der Applesoft-Zeile ab der momentanen Cursor-Position und Übernahme des Zeilenanfangs in den Programmspeicher.
ESC
Verlassen des Editier-Modus, *ohne* daß die (veränderte) Zeile in den Programmspeicher übernommen wird. Ctrl-X und Ctrl-C bewirken das gleiche.

K
Wie **Z**, doch wird die Applesoft-Zeile in gepackter Form, d.h. ohne überflüssige Leertasten, angezeigt. Nützlich bei überlangen Programmzeilen.

Ctrl-K
Ausgabe des Makros „CATALOG".
Ctrl-L
Ausgabe des Makros „LIST".
Diese beiden Makros sind vordefiniert.

Ctrl-D
Es erscheint das Wort „Macro", und man kann nunmehr ein eigenes Tastatur-Makro definieren. Dazu tippt man zunächst Ctrl-Q oder Ctrl-W oder Ctrl-E oder Ctrl-R oder Ctrl-T oder Ctrl-P (alle in der dritten Reihe der Tastatur) und daraufhin das Makro (z.B. HOME: LIST usw.). Ein Makro kann 100 Zeichen lang sein. Das definierte Makro kann dann mit Ctrl-Q usw. aufgerufen werden. Wenn man bestimmte Makros permanent benötigt, so speichere man den ganzen Editor mit
BSAVE EDITOR.NEU, A32000, L6399

M + Monitorbefehl
führt einen Monitorbefehl von BASIC aus durch, z.B. M300L entspricht CALL -151, 300L, Ctrl-C.

&D + Dezimalzahl
verwandelt eine Dezimalzahl in eine Hexzahl, z.B. &D16 ergibt $0010.
&H + Hexzahl
verwandelt eine Hexzahl in eine Dezimalzahl, z.B. &H10 ergibt 16.

&L
zeigt die Länge des Applesoft-Programms in Bytes an.
&F
zeigt die Anzahl der freien Sektoren auf der Diskette an, auf die man zuvor mit Ctrl-K o.ä. zugreifen sollte; funktioniert nicht korrekt bei bestimmten, für 40 oder 80 Spuren modifizierten DOS-Varianten.

&V
listet alle Variablen des Applesoft-Programms in alphabetischer Reihenfolge. Mit POKE 38395, A (z.B. POKE 38395, 5) kann man die Anzahl der Verweise pro Zeile für den Ausdruck festlegen.
&R z,i
„renumbert" das gesamte Applesoft-Programm ab Zeile z im Intervall i, z.B. &R100,5

&K
entfernt bzw. „kruncht" alle REMs; vorher kommentiertes Programm abspeichern!

&S
Dieser Befehl dient in der Form
&S: Token, z.B. &S:GOTO
zum Suchen von Tokens und in der Form
&S"String, z.B. &S"HALLO
zum Suchen von Strings. Bei gemischten Ausdrücken muß man im Zweifelsfall beide Varianten durchprobieren.

& + Nummer
zeigt eine Applesoft-Zeile in der internen Form als ASCII-Hexdump an, z.B.
&100

len, daß beim Kopieren von geschützten Disketten mit sog. Nibble-Kopierern immer die ganze Spur kopiert werden kann, empfiehlt es sich, die Umdrehungsdauer auf ca. +0,5 % einzustellen (vorausgesetzt die +1 % Grenze wird dabei durch andere Einflüsse nicht überschritten). Durch die längere Umdrehungsdauer passen mehr Bytes auf eine Spur, was eine zusätzliche Reserve beim Nibble-Kopieren darstellt. Beim normalen Betrieb mit allen Betriebssystemen stört eine Abweichung von 0,5 % nicht im geringsten.

4.2. Spurjustage

Die Überprüfung, ob der Schreib-/Lesekopf exakt auf die vorgeschriebenen Spuren positioniert wird und ob eine eventuelle Korrektur der Spurlage nötig ist, ist relativ schwierig und aufwendig. Zunächst wird ein Oszillograph benötigt, da das analoge Lesesignal gemessen werden muß. Als nächstes wird eine Referenz-Diskette gebraucht, bei der die Aufzeichnung ganz genau auf der genormten Sollspurlage erfolgt ist. Diese Forderung ist nur bei speziellen **Justagedisketten** erfüllt, die von verschiedenen Diskettenherstellern angeboten werden. Wegen der erforderlichen Präzision und den geringen Stückzahlen sind solche Justagedisketten nicht gerade billig (ca. 150.- bis 200.- DM). Da sie aber auch nicht oft benötigt werden, reicht es, wenn in einem Club oder einer Gruppe nur eine Diskette gekauft und dann ausgetauscht wird. Das Kopieren einer Justagediskette ist technisch nicht möglich.

4.2.1. Spurjustage-Theorie

Für die Spurjustage des DISK II Laufwerks ist die Justagediskette von BASF besonders geeignet und wurde deshalb auch für diesen Artikel verwendet. Die Justagediskette hat die Bezeichnung „BASF CE-FlexyDisk 5.25" (CE = Customer Engineering = Kundendienst) und ist in vier Versionen erhältlich: einseitig-einfache Dichte, doppelseitig-einfache Dichte, einseitig-doppelte Dichte und doppelseitig-doppelte Dichte. Für die Justage des DISK II und kompatibler Laufwerke reicht die Version für einseitige Aufzeichnung und einfache Dichte aus. Die BASF-Justagediskette kann von folgenden Firmen bezogen werden:
Fa. Feltron Elektronik, 5210 Troisdorf
Fa. Emil Löffelhardt, 7012 Fellbach
Für die Spurjustage enthält die Justagediskette die in **Bild 9** gezeigte Aufzeichnung. Links im Bild ist die Sollspurlage für die Spuren 15, 16 und 17 gezeigt, rechts daneben das Aufzeichnungsformat eines

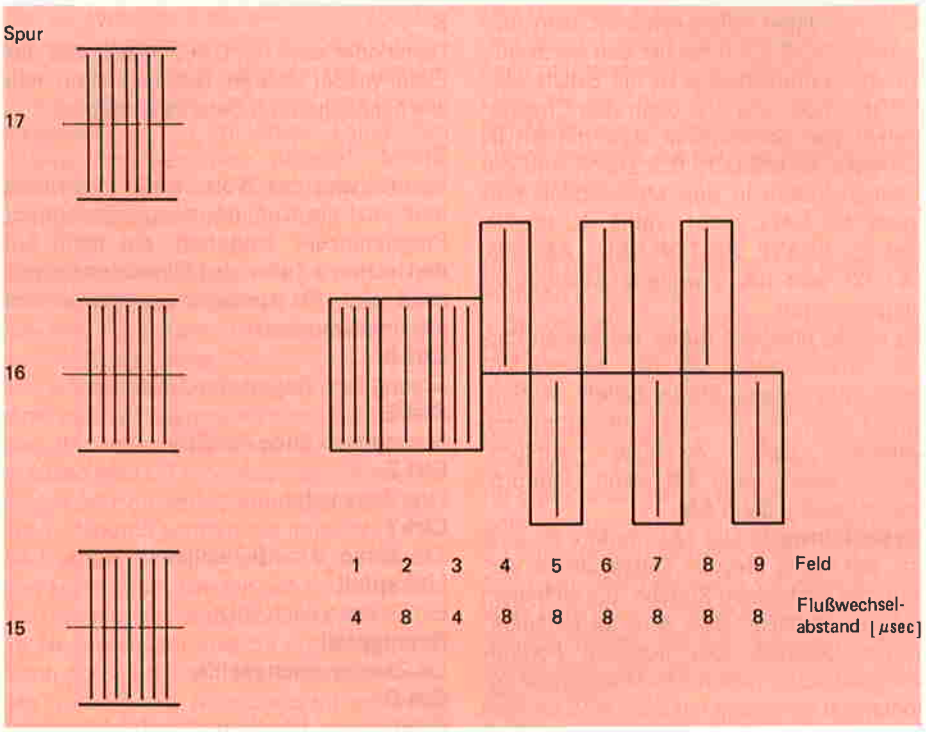


Bild 9: Spurjustagediskette-Aufzeichnungsformate

Sektors auf der Justagediskette. Danach besteht ein Sektor aus 9 gleichlangen Feldern. Die ersten drei Felder dienen lediglich zur Orientierung beim Messen. Sie sind genau auf Sollspurlage von Spur 16, jedoch mit unterschiedlichen Frequenzen aufgezeichnet. Die Aufzeichnung in Feld 1* und 3 hat einen Flußwechselabstand von 4 µsec. Dagegen hat die Aufzeichnung in Feld 2, ebenso wie in den Feldern 4 bis 9, einen Flußwechselabstand von 8 µsec. Zur eigentlichen Justage werden die Felder 4 bis 9 benötigt. Die Besonderheit dieser Felder liegt darin, daß die geradzahligen Felder genau um eine halbe Spurbreite nach innen (in Richtung auf Spur 17) und die ungeradzahligen Felder genau um eine halbe Spurbreite nach außen versetzt sind. Auf der gesamten Spur sind sechs gleiche Sektoren (ohne Zwischenräume) aufgezeichnet, entsprechend 60 Grad pro Sektor.

Bei der Justage wird das analoge Lesesignal dieser Aufzeichnung mit einem Oszillographen gemessen. Dabei sind zwei Effekte zu beobachten:
– Die Lesesignalamplitude bei einer Aufzeichnung mit 4 µsec Flußwechselabstand (Feld 1 und Feld 3) ist kleiner als die bei einer Aufzeichnung mit 8 µsec (Feld 2).
– Bei gleichem Flußwechselabstand ist die Lesesignalamplitude proportional zu der vom Lesekopf erfaßten Spurbreite, d.h. je mehr von der geschriebenen Spur gelesen wird, desto größer ist das Lesesignal.

Das sich aus diesen Effekten ergebende Lesesignal zeigen die Bilder 10 und 11.

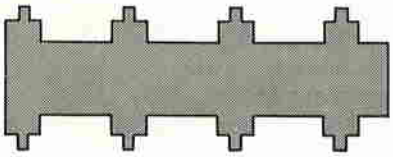


Bild 10: Ideale Spurposition

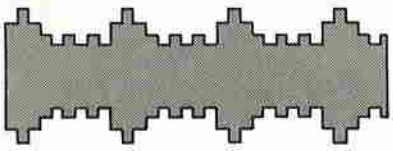


Bild 11: Spurabweichung

* Zu den Feldern siehe Bild 12 auf Seite 18.

Bild 10 zeigt den Idealfall, daß der Lesekopf genau auf der vorgeschriebenen Spurposition (Spur 16) steht. Das Lesesignal in Feld 2 hat die größte Amplitude, da die Aufzeichnung mit 8 µsec Flußwechselabstand erfolgt ist und der Lesekopf die geschriebene Spurbreite von Feld 2 voll erfaßt. Die Amplitude der Felder 1 und 3 ist etwas geringer, da in diesen Feldern die Aufzeichnung mit 4 µsec Flußwechselabstand erfolgt ist. In den Feldern 4 bis 9 ist die Amplitude in allen Feldern gleich groß, aber nur genau halb so groß wie die Amplitude in Feld 2. Dies kommt daher, daß diese Felder bei der Aufzeichnung um eine halbe Spurbreite versetzt aufgezeichnet wurden, und der Lesekopf daher nur jeweils die halbe Spurbreite erfaßt.

Bild 11 zeigt das Lesesignal, wenn der Lesekopf nicht genau auf der vorgeschriebenen Spurposition steht. In diesem Fall hat der Lesekopf eine Spurabweichung von 20 µm, was 40% des maximal zulässigen Fehlers entspricht. Wenn man Bild 9 betrachtet und sich vorstellt, daß der Lesekopf von Spur 16 aus etwas nach oben (in Richtung zur Spur 17) verschoben wird, so ist klar, daß der Lesekopf einen größeren Anteil der geraden Felder (4, 6, 8) und dagegen einen kleineren Anteil der ungeraden Felder (5, 7, 9) erfaßt. Entsprechend steigt in Bild 11 die Lesesignalamplitude der geraden Felder gegenüber Bild 10, während die Lesesignalamplitude der ungeraden Felder sinkt.

Bei einer Spurabweichung in umgekehrter Richtung kehren sich die Verhältnisse genau um, die Amplitude der geraden Felder sinkt, und die Amplitude der ungeraden Felder steigt an.

Aus dem Verhältnis der Amplituden kann man direkt die Spurabweichung (SA) nach folgender Formel ausrechnen:

$$SA = \frac{Ug - Uu}{Ug + Uu} \cdot \frac{SB}{2}$$

In der Gleichung ist Ug die Amplitude (Spitze zu Spitze) eines geraden Feldes und Uu die Amplitude des danebenliegenden ungeraden Feldes. Der Wert SB ist die Spurbreite (300 µm bei 48 tpi und 146 µm bei 96 tpi).

In der Praxis werden die Verhältnisse noch etwas komplizierter, da oft die Justagediskette nicht ganz genau konzentrisch eingespannt wird (es geht hier nur um einige Mikrometer!). Durch eine exzentrische Einspannung ergibt sich aber, daß der Lesekopf in jedem Feld einen etwas anderen Spurbereich erfaßt und damit in allen Feldern unterschiedliche Amplituden gemessen werden. Der Fehler einer exzentrischen Einspannung kann bei der BASF Justagediskette ganz leicht eliminiert werden. Man braucht nur die Spurabweichung für zwei gegenüberliegende Felder (SA1 und SA4) zu bestimmen und aus ihnen den Mittelwert (SAM) zu bilden:

```
SAM = (SA1 + SA4) / 2
```

Da jeder Sektor 60 Grad lang ist, liegen sich Felder dann gegenüber, wenn sie die gleiche Feldnummer haben und sich die Sektornummern um drei unterscheiden (z.B. liegen sich Feld 4 in Sektor 1 und Feld 4 in Sektor 4 gegenüber). Bei der Mittelwertbildung muß das Vorzeichen immer mit berücksichtigt werden.

4.2.2. Spurjustage-Simulation

Um sich die Vorgänge bei der Spurjustage leichter vorstellen zu können, wurde ein Simulationsprogramm geschrieben. Das Programm simuliert die Messung der Spurabweichung mit der Justagediskette und gibt auf dem Bildschirm das Bild aus, das man bei der Messung auf dem Oszillographen sehen würde. Wenn man für die Spurabweichung unterschiedliche Werte eingibt, sieht man sehr leicht, wie die verschiedenen Oszillographenbilder bei der Justage zustande kommen. Die Bilder 10 bis 12 wurden übrigens auch mit diesem Programm generiert.

Das Simulationsprogramm, das auf der Peeker Sammeldiskette #22 enthalten ist, wird von Diskette mit dem Befehl „RUN SPURJUSTAGE.SIMULATION" gestartet. Nach einer kurzen Zeit zur Initialisierung des Programms müssen Spurabweichung und Exzentrizität eingegeben werden. Beide Werte werden in Mikrometer eingegeben und dürfen den Wert 500 nicht überschreiten (dies entspricht dem Abstand zwischen zwei Spuren bei 48 tpi). Wenn die Exzentrizität ungleich Null ist, so muß auch noch ein Winkel eingegeben werden. Dieser Winkel, der einen Wert zwischen 0 und 360 (Grad) haben darf, definiert den Abstand zwischen dem simuliertem Triggerpunkt (dem Anfang der Kurven) und dem Maximum der Exzentrizität.
In der oberen Hälfte des Bildschirms wird die Spurlage der einzelnen Felder dargestellt. Darin eingeblendet wird die Bahn, die der Lesekopf relativ zur Aufzeichnung bestreicht. In der unteren Hälfte des Bildschirms wird das Oszillographenbild simuliert, das man bei der Justage sieht.
Nachdem die Bilder gezeichnet wurden, kann zwischen den folgenden Funktionen ausgewählt werden:

```
1 - Darstellung der HGR Seite 1
      auf dem Bildschirm
2 - Darstellung der HGR Seite 2
      auf dem Bildschirm
K - Kopieren der HGR Seite 1 in
      die HGR Seite 2
N - Ausgabe eines neuen Bildes
      (erfolgt immer in HGR Seite 1)
D - Ausdrucken des momentan dar-
      gestellten Bildes
B - Rückkehr nach BASIC
```

Nach Anwahl der Funktion 1 oder 2 wird außerdem unter den beiden Bildern noch die Aufteilung in die sechs Sektoren gezeigt.

Mit den verfügbaren Funktionen kann man z.B. zuerst ein Bild in Sollspurlage ausgeben und von der HGR Seite 1 in die Seite 2 kopieren. Danach gibt man ein Bild mit einer Spurabweichung aus. Mit den Tasten „1" und „2" kann man dann sehr schnell zwischen den beiden HGR Seiten hin- und herschalten und sieht so, wie sich das Oszillographenbild ändern würde, wenn man den Lesekopf geringfügig aus der Sollspurlage verschiebt.

Wem das Programm zu langsam läuft, der kann es ohne Änderungen mit dem TASC-Compiler compilieren. Dabei darf jedoch nicht die Standard-Speicherkonfiguration benutzt werden, sondern als Programm-Adresse muß „HGR2" eingegeben werden. Die Warnung während des Compilierens, daß die Zeile 1120 ignoriert wird, hat keine Bedeutung, da der LOMEM-Befehl nur benötigt wird, wenn das Programm vom Interpreter ausgeführt wird.

4.2.3. Überprüfung der Spurlage

Bevor mit der Messung begonnen werden kann, muß der erforderliche Meßaufbau zusammengestellt werden. Der benötigte Oszillograph muß folgendermaßen angeschlossen und eingestellt werden:
– Y-Eingang: Anschluß eines 1:1 Tastkopfes an Testpunkt TP8 oder TP9 (s. Bild 1). Masse des Tastkopfes an Testpunkt TP3. (Wenn der verwendete Oszillograph zwei Y-Eingänge hat und die Möglichkeit besteht, die beiden Kanäle voneinander abzuziehen (einen Kanal invertieren und die Kanäle addieren), ist es günstiger, die beiden Eingänge an die Testpunkte TP8 und TP9 anzuschließen und die Differenz der beiden Signale zu messen.)
– Kopplung: AC (Wechselspannung)
– Eingangsempfindlichkeit: 50 Millivolt pro Skalenteil. (Wenn der verwendete Oszillograph eine Eingangsempfindlichkeit von 5 mV pro Teilung hat, so ist es günstiger, diese Empfindlichkeit zu wählen und anstatt der 1:1 Tastköpfe 10:1 Tastköpfe zu verwenden.)
– Zeitablenkung: 20 msec pro Skalenteil.

– Trigger: Externer Trigger – Anschluß an den Utility-Strobe-Ausgang des Apple (Game I/O Stecker Stift 5). (Bei Laufwerken, die über eine Index-Erkennung verfügen, kann statt dessen auch das Index-Signal benutzt werden. Bei der 34 poligen „Shugart-Schnittstelle" ist dies Anschluß 8.)

Zur Überprüfung aller Anschlüsse und zur optimalen Einstellung des Oszillographen sollte zunächst mit einer normalen (leeren!) Diskette „geübt" werden, um ein versehentliches Löschen der teuren Justagediskette zu vermeiden. Mit Hilfe des DISKTEST-Programms wird das zu justierende Laufwerk auf Spur 16 positioniert und diese Spur mit einem Flußwechselabstand von 4 µsec beschrieben. Nachdem dann der J-Befehl gegeben wurde, muß der Lautsprecher des Apple ein periodisches „Klicken" abgeben und der Oszillograph ein Triggersignal bekommen. Der Oszillograph muß dann ein breites Band mit einer Amplitude von ca. 300 mVss zeigen. (Wegen der niedrigen Ablenkgeschwindigkeit sieht man nicht die einzelnen Schwingungen des Lesesignals.) Helligkeit, Trigger, Horizontal- und Vertikalposition können mit der Übungsdiskette in aller Ruhe auf optimale Werte eingestellt werden.

Das Laufwerk kann jetzt ausgeschaltet und die Justage-Diskette eingelegt werden. Nach Wiedereinschalten des Laufwerks und erneutem J-Befehl muß der Oszillograph einen dem Bild 10 oder Bild 11 ähnlichen Signalverlauf zeigen. Da normalerweise kein Strom durch den Schrittmotor fließt, wird der Schreib-/Lesekopf nicht in seiner Position festgehalten, und man kann durch vorsichtiges Drehen an der Spiralscheibe den Kopf in unterschiedliche Positionen bringen.

Bei Triggerung mit dem DISKTEST-Programm beginnt das Oszillographenbild immer mit den **Orientierungsfeldern** (Feld 1 bis 3). Man erkennt die Orientierungsfelder sehr leicht daran, daß die drei Felder immer gleichsinnig in der Amplitude zu- oder abnehmen, wenn der Kopf bewegt wird, und Feld 2 immer eine größere Amplitude hat als die Felder 1 und 3. Wenn die Felder 1 bis 3 die größte Amplitude haben, steht der Kopf auf Sollspurlage. Da die Einstellung auf ein Maximum aber nie sehr genau ist (z.B. auch in der Funkpeilung), werden zur Justage die Felder 4 und 5 benutzt. Wenn man die Kopfposition leicht verändert, sieht man, daß sich die Amplitude der beiden Felder gegenläufig ändert. Wenn die Amplitude von Feld 4 zunimmt, nimmt die von Feld 5 ab und umgekehrt. Wenn beide Amplituden gleich groß sind, ist der Kopf wieder auf Sollspurlage. Man merkt jedoch sehr deutlich, daß die Einstellung mit den Feldern 4 und 5 viel genauer ist als mit den Feldern 1 bis 3.

Zur Überprüfung der Spurlage ist es nicht erforderlich, die absolute Spurabweichung auszurechnen, sondern es reicht festzustellen, ob die Abweichung innerhalb einer vorgeschriebenen Toleranzgrenze liegt. Deshalb braucht man auch nicht die oben angegebene Formel zu benutzen, sondern kann ein einfacheres Verfahren verwenden.

Vor Beginn der Messung muß zunächst der Kopf wieder mit dem Schrittmotor auf eine definierte Position gebracht werden, falls er von Hand verschoben wurde. Dazu wird am besten der Kopf zuerst mit dem 0-Befehl auf Spur 0 und dann mit dem P-Befehl wieder auf Spur 16 positioniert.
Wenn der Kopf auf der richtigen Position steht und der Trigger mit dem J-Befehl wieder eingeschaltet wurde, muß der Oszillograph richtig eingestellt werden. Die Zeitablenkung sollte jetzt auf 10 msec eingestellt werden und mit der variablen Zeitbasis so, daß – wie in Bild 12 gezeigt – ca. 3 1/2 Sektoren auf dem Schirm sichtbar sind. Dadurch wird erreicht, daß der Oszillograph bei jeder Umdrehung getriggert wird, was ein besseres Bild zur Folge hat.

Danach werden aus dem Schirmbild die in **Bild 12** markierten Amplitudenwerte U14 und U15 (Sektor 1 Feld 4 und 5) durcheinander geteilt. Ein Wert zwischen 0,5 und 2 – das entspricht einer Spurabweichung von maximal 50 µm – bildet die Toleranzgrenze für die Spurlage.

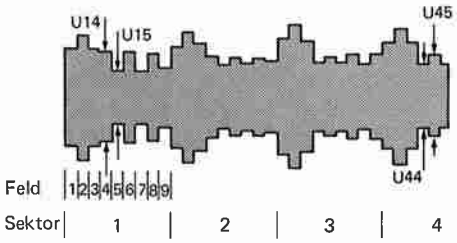


Bild 12: Spurposition – Oszillographenbild

Zum genauen Ablesen verschiebt man am besten die horizontale Position des Oszillographenbildes so, daß sich das jeweilige Feld auf der vertikalen Skala in der Mitte des Schirmes befindet. Wenn der verwendete Oszillograph eine variable Y-Verstärkung hat, stellt man am besten die Amplitude des größeren Feldes (4 oder 5) auf einen geraden Skalenwert (z.B. 4 Skalenteile) ein. Das kleinere der beiden Felder muß jetzt mindestens die halbe Amplitude (z.B. 2 Skalenteile) haben. Wenn der Oszillograph keine variable Y-Verstärkung hat, muß man die Werte für U14 und U15 ablesen und durcheinander teilen. Da sich bei der Teilung die Einheiten herauskürzen, ist es nicht erforderlich, die Spannungen in Volt auszurechnen, sondern man kann vielmehr direkt in Skalenteilen rechnen. Das Ergebnis der Rechnung muß zwischen 0,5 und 2 liegen.

Wenn das Verhältnis zwischen geraden und ungeraden Feldern in allen Sektoren gleich ist (d.h. wenn die Diskette nicht exzentrisch eingespannt ist), sind keine weiteren Messungen erforderlich. Wenn eine der obigen Bedingungen, die aus den Justagevorschriften der BASF abgeleitet wurden, eingehalten wird, ist die Spurlage des Laufwerks genau genug und eine Korrektur der Justage nicht erforderlich. Da von der Firma Apple keine Informationen zu erhalten waren, sollte man für das Apple Laufwerk wohl von den gleichen Werten ausgehen.

Die Überprüfung der Spurlage wird etwas schwieriger, wenn nicht – wie in Bild 11 gezeigt – das Verhältnis zwischen geraden und ungeraden Feldern in allen Sektoren gleich ist. Bild 12 zeigt einen solchen Fall, bei dem sich das Verhältnis von Feld zu Feld ändert. Dies kommt daher, daß die Diskette nicht genau konzentrisch eingespannt war (30 µm Exzentrizität und 5 µm Abweichung). Eine Exzentrizität kann auftreten, wenn das Innenloch der Diskette etwas größer als die Aufnahme des Laufwerks ist, wenn die Diskette bei der Einspannung nicht genau zentriert wird oder wenn die Spindel des Laufwerks nicht exakt rund läuft. Wenn dies der Fall ist, sollte man zunächst versuchen, durch mehrmaliges Öffnen und Schließen der Frontklappe (ggf. bei laufendem Motor) eine konzentrische Einspannung der Diskette zu erreichen.

Wenn dies nicht gelingt, muß man die obige Prüfung in den Sektoren 1 und 4 nacheinander machen. Wenn in beiden Sektoren die angegebene Bedingung erfüllt wird, ist die Spurlage auch in Ordnung. Wenn jedoch in einem oder beiden Sektoren die Bedingung nicht erfüllt wird, muß folgende Rechnung gemacht werden:

$$X = \frac{U14 \cdot U44}{U15 \cdot U45}$$

Durch diese Rechnung wird der Fehler ausgeglichen, der sich aus einer exzentrischen Einspannung ergibt. Wenn der Wert für X zwischen 0,25 und 4 liegt, ist die Spurlage auch in Ordnung.

Bei der Überprüfung der Spurlage muß noch ein weiterer Effekt beachtet werden, nämlich die **Hysterese** des Positioniersystems. Wegen der mechanischen Reibung innerhalb des Positioniersystems ist die genaue Kopfposition etwas unterschiedlich, je nachdem, aus welcher Richtung der Kopf auf die gewünschte Position fährt. Deshalb sollte man bei der Überprüfung der Spurlage (und bei einer eventuellen Justage) zwei Messungen machen. Vor der einen Messung sollte der Kopf mit dem A-Befehl auf Spur 15 und dann mit dem I-Befehl wieder auf Spur 16 positioniert werden. Vor der anderen Messung wird der Kopf zuerst mit dem I-Befehl auf Spur 17 und dann mit dem A-Befehl wieder auf Spur 16 positioniert. Dadurch wird erreicht, daß der Kopf für beide Messungen aus unterschiedlicher Richtung auf Spur 16 fährt. In beiden Fällen soll die Spurabweichung innerhalb der vorgeschriebenen Toleranz liegen.

4.2.4. Korrektur der Spurlage

Die Spurjustage ist ziemlich schwierig und erfordert einige Geduld und Übung. Sie sollte deshalb nur vorgenommen werden, wenn die Spurlage eindeutig außerhalb der vorgeschriebenen Toleranz liegt.

Zur Korrektur der Spurlage werden die zwei Schrauben gelöst, die den Schrittmotor halten. Durch Drehen des Schrittmotors kann dann die genaue Spurposition beliebig verändert werden. Dabei ist zu beachten, daß beim Verdrehen des Motors der Kopf nicht entsprechend mitbewegt wird, da der Schrittmotor nicht ständig bestromt wird. Vielmehr werden kurz nach jedem Positionieren alle Phasen wieder abgeschaltet. Nach Verdrehen des Motors muß der Kopf also erst durch erneutes Positionieren (auf Spur 0 und dann Spur 16) auf die vom Schrittmotor kontrollierte Position gebracht werden. Um das ständige erneute Positionieren nach jeder Verdrehung des Motors zu vermeiden, ist es auch möglich, das DISKTEST-Programm durch Einfügen von 10300 POKE 37115,96 (s. Abschnitt 5) so zu ändern, daß der Schrittmotor ständig bestromt ist und der Kopf bei jeder Drehung des Schrittmotors entsprechend mitbewegt wird.

Zur Justage wird der Schrittmotor einfach so lange verdreht, bis der Kopf die Sollspurlage erreicht hat. Die Justage sollte so genau wie möglich vorgenommen werden, d.h. die geraden und ungeraden Felder sollten möglichst gleich sein. Auf jeden Fall soll die Justage genauer als die Überprüfung sein. Wenn das größere der beiden Felder 4 Skalenteile groß ist, dann sollte das kleinere mindestens 3,5 Skalenteile groß sein, was einer Abweichung von maximal 10 µm entspricht. Wenn dies erreicht ist, muß der Schrittmotor festgeschraubt und damit die Justage fixiert werden.

Dieses Verfahren hat zwei Nachteile. Erstens verdreht sich der Schrittmotor beim Festschrauben meist noch geringfügig und zweitens entspricht die beim Verdrehen des Schrittmotors erreichte Position nicht genau der nach dem Positionieren erreichten Position. Meist geht die Justage mit dem folgenden Verfahren schneller und genauer:

- Schrittmotor lösen
- Schrittmotor verdrehen
- Schrittmotor festschrauben
- neu positionieren
- Spurlage überprüfen

Nach einigen Versuchen (ggf. mit Markierungen am Gehäuse und den Schrittmotorflanschen) bekommt man ein Gefühl, um wieviel und in welcher Richtung man den Schrittmotor drehen muß, um auf die richtige Position zu kommen.

5. Tips zur Fehlersuche

In diesem Abschnitt sollen einige Tips zur Fehlersuche gegeben werden, für den Fall, daß das Disketten-System einmal seinen Dienst versagt.
Als erstes sollte durch Benutzung einer anderen DOS-Diskette, durch Verwendung eines anderen Controllers und Anschluß eines anderen Disketten-Laufwerks sichergestellt sein, daß der Fehler wirklich im Laufwerk und nicht in einer anderen Komponente liegt.
Bevor das Laufwerk für die nachfolgenden Untersuchungen geöffnet wird, sollte dann noch der Aufbau des Systems untersucht werden. Manche Laufwerke sind nämlich sehr empfindlich gegen magnetische Einstreuungen. Bei solchen Laufwerken treten häufig Lesefehler auf, wenn sie in unmittelbarer Nähe des Monitors (Fernsehers) betrieben werden. Wenn das Laufwerk etwas von der Störquelle entfernt wird, sind diese Fehler mit einem Schlag verschwunden.
Wenn die Maßnahmen bis dahin nicht erfolgreich waren, ist es für die weiteren Untersuchungen erforderlich, das Laufwerk zu öffnen (siehe Abschnitt 1).
Nach Öffnen des Laufwerks sollte als erstes nachgemessen werden, ob alle Versorgungsspannungen vorhanden sind und innerhalb der angegebenen Grenzen liegen:

```
+ 5 V:  + 4,75 V bis + 5,25 V
+12 V:  +10,80 V bis +13,20 V
-12 V:  -10,80 V bis -13,20 V
```

Die Messung erfolgt am besten jeweils an den beiden Anschlüssen der Kondensatoren C4 (+5 V), C2 (+12 V) und C5 (-12 V). Die angegebenen Grenzen sollen sowohl bei ein- als auch bei ausgeschaltetem Laufwerk nicht überschritten werden. Da der größte Stromverbrauch beim Anlaufen des Motors auftritt, sollte durch einen besonderen Test überprüft werden, ob die +12 V-Spannung nicht beim Anlaufen des Motors zusammenbricht. Hierzu dient das Programm MOTORTEST, das das Laufwerk ständig ein- und ausschaltet. Da Meßinstrumente im allgemeinen zu träge sind, um kurzzeitige Spannungseinbrüche zu messen, sollte zur Messung ein Oszillograph verwendet werden. Als Trigger erzeugt das Programm einen Impuls am Utility-Strobe-Ausgang (Game I/O Stecker Stift 5).

Das MOTORTEST-Programm erzeugt in Zeile 180 zuerst den Triggerimpuls und dann einen kurzen Impuls zum Einschalten des Laufwerks. Die Einschaltdauer wird durch ein Monoflop auf der Controllerkarte bestimmt. Durch Exemplarstreuungen kann diese Zeit unterschiedlich groß sein. Zum Ausgleich kann die Wartezeit in Zeile 190 so verändert werden, daß Ein- und Ausschaltdauer ungefähr gleich groß sind.

Wenn bei einem System alle Slots bestückt sind und das Netzteil zu schwach ist, können eine oder mehrere Spannungen unter die angegebene Grenze absinken, was zu Fehlern des Diskettenlaufwerkes und/oder anderer Systemkomponenten führen kann. Dies tritt insbesondere bei Verwendung von Nicht-Apple-Laufwerken auf, da diese oft einen höheren Stromverbrauch als Original-Apple-Laufwerke haben. In einem solchen Fall muß entweder ein stärkeres Netzteil verwendet werden oder die Diskettenlaufwerke müssen aus einem separaten Netzteil gespeist werden.

Für die weiteren Untersuchungen sollte das DISKTEST-Programm benutzt werden, das von einem zweiten Laufwerk oder Kassette geladen werden kann.

Wenn das Laufwerk (mit dem E-Befehl) eingeschaltet wird, muß die Betriebsanzeige-LED aufleuchten. Tut sie das nicht, so ist wahrscheinlich der IC ULN2003 oder der Transistor T1 defekt.

Mit dem Einschalten des Laufwerks muß gleichzeitig auch der Disketten-Antriebsmotor anlaufen. Ist dies nicht der Fall (und die Diskette nicht verklemmt), dann ist der Fehler auf der Motorregelungs-Karte zu suchen.

Läuft der Motor, sollte als nächstes die Drehzahl überprüft werden (siehe Abschnitt 4.1). Wenn die Überprüfung mit dem DISKTEST-Programm (wegen eines Fehlers im Schreib- oder Lesekreis) nicht möglich ist, ist zunächst eine grobe Überprüfung mit der auf der großen Riemenscheibe aufgeklebten Stroboskopscheibe möglich. Wenn die Stroboskopscheibe mit künstlichem Licht (am besten mit einer Leuchtstoffröhre) beleuchtet wird, so dürfen sich bei laufendem Motor die Striche des inneren (mit „50" bezeichneten) Rings scheinbar gar nicht oder nur ganz langsam bewegen. (Nach Reparatur des Schreib-/Lesekreises sollte die Drehzahlüberprüfung bzw. Justage auf jeden Fall aber noch einmal mit dem DISKTEST-Programm wiederholt werden.)

Wenn der Diskettenantrieb in Ordnung ist, kann als nächstes die Positionierung untersucht werden. Dazu läßt man am besten das Laufwerk einige Zeit mit dem H-Befehl zwischen den äußersten Spuren (0 und 34) hin- und herpositionieren. Wenn der Kopfschlitten sich gar nicht bewegt, ungleichmäßig läuft, anstößt oder nicht den ganzen Aufzeichnungsbereich überstreicht, ist ein Defekt der Treiber im IC ULN 2003 wahrscheinlich. Um dies zu überprüfen, werden die einzelnen Phasen nacheinander angesteuert und gemessen. Mit dem DISKTEST-Programm ist das nicht ganz einfach, da schon kurz nach jeder Positionierung alle Phasen wieder abgeschaltet werden. Durch die folgende Änderung des DISKTEST-Programms wird dies vermieden:

```
10300 PH=1 : POKE 37115,96
```

Durch den Befehl PH=1 wird die Positionierung auf Halbspur-Betrieb umgestellt. Die Änderung des Speicherplatzes 37115 bewirkt, daß die Phasen nach dem Positionieren nicht mehr abgeschaltet werden. Mit den Befehlen I und A wird jetzt eine Phase nach der anderen eingeschaltet, und das Ergebnis kann einfach mit einem Voltmeter an den verschiedenen Stellen der Schaltung nachgemessen werden. Bei diesem Verfahren muß jedoch darauf geachtet werden, daß sich der Treiberbaustein ULN 2003 und der Schrittmotor nicht zu stark erwärmen. Die geänderte Version sollte deshalb vorsichtshalber auch nicht auf Diskette gespeichert werden, um zu vermeiden, daß das Laufwerk versehentlich längere Zeit mit dem geänderten Programm betrieben wird.

Quellen

Apple Computer Inc., Apple II, The DOS Manual, Disk Operating System
BASF 6106/6108, Mini Disk Drive, Technical Manual, Part No. 80307-050
BASF AG, Specification and Description, BASF CE-FlexyDisk 5,25 130 mm for track adjustment, amplitude control and index sensor alignment
SGS, Datenblatt ULN2003A
SONY, Datenblatt CX-065B

SPURJUSTAGE.SIMULATION

```
1000  REM Disketten Spurjustage Simulation
1010  REM von Dipl.-Ing. Gerhard Berg
1020  REM 31.3.1986
1030  :
1100  IF PEEK (104) > 95 GOTO 1130
1110  IF PEEK (176) > 31 THEN STOP
1120  LOMEM: 24576
1130  GOSUB 2000
1140  :
1200  FOR X = 0 TO 269
1210  KO = KL-INT (EX * CO(X + WI) + 0.5)
1220  SO = SL(X):D = KO - SO:KU = KO + SB
1230  A = INT ((SB - ABS (D)) * A(X))
1240  IF A < 0 THEN A = 0
1250  HPLOT X,SO TO X,SO + SB
1260  IF D = 0 OR KU < 0 OR KO > 79 GOTO 1290
1270  IF KO >= 0 THEN XDRAW 1 AT X,KO
1280  IF KU < 80 THEN XDRAW 1 AT X,KU
1290  HPLOT X,N2 - A TO X,N2 + A: NEXT
1300  FOR X = 0 TO 270 STEP 45
1310  HPLOT X,170 TO X,180: NEXT
1320  :
1500  PRINT "1/2=Bild #; K=Kopieren; N=Neues Bild: ";
1510  B$ = "12KNDB": GET E$
1520  FOR B = 1 TO 6 IF E$ = MID$ (B$,B,1) GOTO 1540
1530  NEXT: CALL -198: GOTO 1510
1540  ON B GOTO 1550,1550,1560,1590,1600,1680
1550  POKE HM,0:S = B: POKE HS + S - 1,0 GOTO 1510
1560  POKE HS,0: POKE 60,0: POKE 61,32
1570  POKE 62,255: POKE 63,63: POKE 66,0: POKE 67,64
1580  CALL 774:S = 2: GOSUB 3110:S = 1: GOTO 1510
1590  GOSUB 3000: GOTO 1200
1600  HOME: PRINT
1610  PRINT CHR$ (4); "PR#1": PRINT CHR$ (9); "80N"
1620  PRINT CHR$ (9); "G";: IF S = 2 THEN  PRINT "2";
1630  PRINT: PRINT: PRINT
1640  PRINT "Spurabweichung: "; SA(S) * 10; " micrometer"
1650  PRINT "Exzentrizität:  "; EX(S) * 10; " micrometer"
1660  IF EX(S) <> 0 THEN PRINT "Winkel: "; EW(S); " Grad"
1670  PRINT CHR$ (4); "PR#0": VTAB 21: GOTO 1500
1680  TEXT: HOME: END
1690  :
2000  TEXT: HOME: SPEED= 100: HTAB 4: INVERSE
2010  PRINT "DISKETTEN SPURJUSTAGE SIMULATION"
2020  NORMAL: PRINT: HTAB 6
2030  PRINT "von Gerhard Berg - 31.3.1986": SPEED = 255
2040  VTAB 13: HTAB 13: PRINT "Bitte warten!"
2050  FOR AD = 768 TO 779: READ DA: POKE AD,DA: NEXT
2060  DATA 1,0,4,0,4,0,216,160,0,76,44,254
2070  POKE 232,0: POKE 233,3: REM Shape Tabelle
2080  DIM CO(540),SL(270),A(270)
2090  P1 = 8 * ATN (1) / 270: SB = 30: N1 = 25: N2 = 120
2100  DEF FN IR(Z) = INT (Z / 10 + 0.5 * SGN (Z))
2110  FOR X = 0 TO 67: CO = COS (X * P1)
2120  CO(X) = CO: CO(135 - X) = -CO
2130  CO(135 + X) = -CO: CO(270 - X) = CO: NEXT
2140  FOR X = 0 TO 270: CO(X + 270) = CO(X): NEXT
2150  DATA 0,0.8,0,1,0,0.8,1,1,-1,1,1,1,-1,1,1,1,-1,1
2160  FOR FE = 0 TO 40 STEP 5: READ SL,A
2170  SL = N1 - SL * SB / 2
2180  FOR SE = 0 TO 225 STEP 45
2190  FOR X = SE + FE TO SE + FE + 4
2200  SL(X) = SL: A(X) = A: NEXT: NEXT: NEXT
2210  HE = -16304: HM = -16302: HS = -16300
2220  HCOLOR = 3: ROT = 0: SCALE = 1: POKE -16297,0
2230  POKE 230,64: CALL -3086: REM HGR2 löschen
2240  :
3000  HOME: POKE HS,0: POKE HM + 1,0: VTAB 21
3010  INPUT "Spurabweichung (micrometer): "; SA
3020  SA = FN IR(SA): IF ABS (SA) > 50 GOTO 3010
3030  INPUT "Exzentrizität (micrometer): ";EX
3040  EX = FN IR(EX): IF ABS (EX) > 50 GOTO 3030
3050  IF EX = 0 THEN WI = 0: GOTO 3080
3060  INPUT "Winkel (Grad): ";EW
3070  IF EW < 0 OR EW > 360 GOTO 3060
3080  KL = N1 - SA: WI = INT ((360 - EW) * 270 / 360)
3090  POKE HE + 1,0: POKE 230,32: CALL -3086
3100  POKE HE,0: S = 1
3110  SA(S) = SA: EX(S) = EX: EW(S) = EW
3120  RETURN
```

MOTORTEST

```
100  REM Diskettenlaufwerk-Motortest
110  INPUT "Steckplatz: ";SN
120  IF SN < 1 OR SN > 7 GOTO 110
130  INPUT "Laufwerk: ";LN
140  IF LN < 1 OR LN > 2 GOTO 130
150  SA = 12 * 4096 + (8 + SN) * 16: REM Steckplatz-Adresse
160  HV = PEEK (SA + 9 + LN): REM Laufwerk auswählen
170  REM Trigger ausgeben und Laufwerk einschalten
180  HV = PEEK (49216) + PEEK (SA + 9) + PEEK (SA + 8)
190  FOR I = 0 TO 3000: NEXT : GOTO 180
```

Tabelle 1: Controller-Schnittstelle

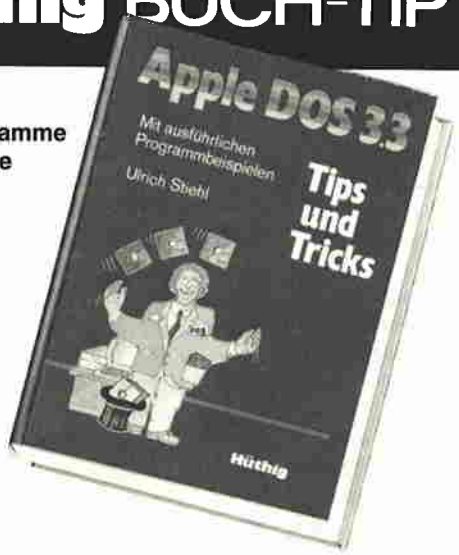
Kontakt	Signal	Funktion (englisch)	Funktion (deutsch)
1	GND	Ground	Masse
2	PhiØ	Phase Ø	Phase Ø
3	GND	Ground	Masse
4	Phi1	Phase 1	Phase 1
5	GND	Ground	Masse
6	Phi2	Phase 2	Phase 2
7	GND	Ground	Masse
8	Phi3	Phase 3	Phase 3
9	-12	-12 Volt	-12 Volt
1Ø	-WR REQ	Write Request	Schreib-Befehl
11	Vcc	+5 Volt	+5 Volt
12	Vcc	+5 Volt	+5 Volt
13	+12	+12 Volt	+12 Volt
14	-ENBL	Enable	Freigabe
15	+12	+12 Volt	+12 Volt
16	RD DATA	Read Data	Lese-Daten
17	+12	+12 Volt	+12 Volt
18	WR DATA	Write Data	Schreib-Daten
19	+12	+12 Volt	+12 Volt
2Ø	W PROT	Write Protect	Schreibschutz

Tabelle 2: Mechanik-Schnittstelle

Kontakt	Stecker	Signal	Funktion (englisch)	Funktion (deutsch)
A	---	---	---	---
1	---	---	---	---
B	rot	ACT LED	Activity LED	Betriebsanzeige
2	schwarz	GND	Ground	Masse (Betriebsanzeige)
C	Klammer	---	---	---
3	Klammer	---	---	---
D	schwarz	W PROT	Write Protect	Schreibschutz-Schalter
4	braun	W P RET	Write Protect Return	Schreibschutz-Schalter Rückleitung
E	---	---	---	---
5	---	GND	Ground	Masse
F	---	---	---	---
6	---	GND	Ground	Masse
H	---	---	---	---
7	---	---	---	---
J	---	---	---	---
8	---	GND	Ground	Masse
K	orange	-MTR ON	Motor On	Motor ein
9	---	---	---	---
L	---	---	---	---
1Ø	---	+12V - ENBL	+12 Volt	+12 Volt (bei Freigabe)
M	braun	PhiC	Phase C	Phase C
11	rot	+12V - ENBL	+12 Volt	+12 Volt (PhiA und PhiC)
N	orange	PhiD	Phase D	Phase D
12	rot	+12V - ENBL	+12 Volt	+12 Volt (PhiB und PhiD)
P	schwarz	PhiA	Phase A	Phase A
13	braun	+12V	+12 Volt	+12 Volt (Motorregelung)
R	gelb	PhiB	Phase B	Phase B
14	schwarz	GND	Ground	Masse (Motorregelung)
S	Klammer	---	---	---
15	Klammer	GND	Ground	Masse

Hüthig BUCH-TIP

Wegen der neuen Programme paßt die neue Begleit-diskette nur zur 3. Auflage.



Apple DOS 3.3 — Tips und Tricks

von U. Stiehl
3., völlig überarb. Aufl. 1986, X, 203, mit zahlreichen, ausführlich kommentierten Programmlistings, kart., DM 28,–
Begleitdiskette ebenfalls DM 28,–

Dr. Alfred Hüthig Verlag · Postf. 10 28 69 · 6900 Heidelberg 1

Tabelle 3: Motorregelungs-Schnittstelle

Stift	Farbe (engl.)	Farbe (deutsch)	Signal	Funktion (englisch)	Funktion (deutsch)
BN	brown	braun	+12V	+12 Volt	+12 Volt
O	orange	orange	-MTR ON	Motor On	Motor ein
BK/BU	black	schwarz	GND	Ground	Masse
	blue	blau	GND	Ground (Drive-Motor)	Masse (Antriebsmotor)
G	green	grün	TG1	Tacho-Generator	Tacho-Generator (Anschluß 1)
Y	yellow	gelb	TG2	Tacho-Generator	Tacho-Generator (Anschluß 2)
R	red	rot	MOT	Drive-Motor	Antriebsmotor

Tabelle 4: Schreib-/Lese-Kopf-Schnittstelle

Kontakt	Signal	Funktion (englisch)	Funktion (deutsch)
1	R/W 1	Read/Write-Head 1	Schreib-/Lese-Kopf Anschluß 1
2	---	---	Stecker-Codierung
3	ERASE	Erase Head	Lösch-Kopf
4	CT	Center Tap	Mittelanzapfung
5	R/W 2	Read/Write-Head 2	Schreib-/Lese-Kopf Anschluß 2

Der Change-Befehl

Eine Utility für Applesoft-Programmierer

von Frank Seide

Die Applesoft-Editoren PRODOS.EDITOR unter ProDOS (Hüthig Software Service) und MACROEDITOR unter DOS 3.3 (Peeker-Sameldisk #16; Heft 4/1986, S.13) haben einen Nachteil: Es ist zwar ein Befehl zum *globalen Suchen* (Search-Befehl), aber nicht zum *globalen Ändern* (Change-Befehl) implementiert worden, der nunmehr mit dem nachfolgenden Programm vorgestellt wird und der natürlich nicht nur in Verbindung mit dem PRODOS. EDITOR oder MACROEDITOR, sondern auch separat verwendet werden kann.
Darüber hinaus ist auf Anregung der Peeker-Redaktion eine weitere Utility eingebaut worden, mit deren Hilfe man die lästigen Leertasten eliminieren kann, die sich zwischen REM bzw. DATA und den nachfolgenden ASCII-Folgen gelegentlich einschleichen.

1. Programmstart

Wir beschreiben zunächst die ProDOS-Version des Change-Befehls, da diese weiter unten als Quelltext gelistet ist. Gestartet wird die Befehlserweiterung unter dem BASIC.SYSTEM mit
-CHANGE.REMS oder
BRUN CHANGE.REMS
Wenn zuvor bereits der PRODOS.EDITOR mit
-PRODOS.EDITOR
geladen worden ist, so verschiebt sich CHANGE.REMS in dem RAM-Bereich unterhalb des Editors. Es ergibt sich dann folgende Speicherverteilung:
$7A00-$7DFF: DOS-Puffer ($0400)
$7E00-$83FF: CHANGE.REMS ($0600)
$8400-$99FF: PRODOS.EDITOR ($1600)
CHANGE.REMS belegt damit 1536 Bytes ($0600), während der PRODOS.EDITOR 5632 Bytes ($1600) einnimmt. Das zu bearbeitende Applesoft-Programm darf also nicht zu groß sein. Andernfalls muß man vorübergehend auf den Editor verzichten und nur die Change-Routine laden, die dann den Anfang des ersten ProDOS-Puffers auf $9000 verlegt.
Für die Benutzung von CHANGE.REMS in Verbindung mit dem MACROEDITOR unter DOS 3.3 (→ CHANGE.REMS. DOS.3.3) beachte man die Hinweise am Ende dieses Aufsatzes.

2. Der CHANGE-Befehl

2.1. Zweck

Das folgende Maschinenprogramm gibt dem Applesoft-Programmierer die Möglichkeit, eine Zeichenfolge in seinem Programm zu suchen und durch eine andere Zeichenfolge zu ersetzen, wie dies auch bei Textverarbeitungsprogrammen üblich ist.

2.2. Aufruf

Das gelistete Programm läuft ohne Veränderung nur unter ProDOS, da es als BASIC.SYSTEM-Erweiterung implementiert worden ist. So wird erreicht, daß das Programm durch einfaches Eintippen des Wortes
CHANGE
aufgerufen werden kann, und zwar nur im Direktmodus. Der Befehl kann sogar im Monitor eingegeben werden, jedoch wird nach Ausführung grundsätzlich ins Applesoft zurückgekehrt. Weitere Applesoft-Erweiterungen kann man oft mit der Change-Erweiterung zusammen benutzen. Das ist u.a. davon abhängig, welche Erweiterung als erste gestartet wurde. Für den PRODOS.EDITOR (und sinngemäß für den MACROEDITOR) gilt:
Erst BRUN PRODOS.EDITOR,
dann BRUN CHANGE.REMS.
Die Syntax des CHANGE-Befehls lautet
CHANGE (Zeilenbereich) (Optionen) Delim Alt-String Delim Neu-String (Delim)
Sie ist an die des Change-Befehls im EDASM- oder BIG-MAC-Assembler angelehnt. Wenn man nur das Wort CHANGE eingibt und Return drückt, erscheint ein „C" und der Cursor. Nun sollte der Benutzer den Rest, also ab ‚Zeilenbereich' eingeben.

2.2.1. Zeilenbereich

Der Zeilenbereich gibt wie beim LIST-Befehl den zu durchsuchenden Bereich an. Er kann weggelassen werden. Dann wird das ganze Programm durchsucht (wie bei LIST ohne Parameter). Beispiele:
CHANGE 10-
CHANGE -100
CHANGE 1200-5634
Statt des Bindestrichs kann auch das Komma benutzt werden (wie bei LIST).

2.2.2. Optionen

Die Optionen werden durch sechs Buchstaben – wahlweise Groß- oder Kleinschreibung – in beliebiger Reihenfolge und Anzahl eingegeben. Dadurch werden bestimmte Suchmodi eingeschaltet. Der folgende Change-Befehl enthält alle implementierten Optionen. Werden die Zeichen weggelassen, dann wird der zugehörige Modus eben nicht eingeschaltet:

CHANGE 17-33 LTKUSV '3.1415'PI'

Die Bedeutungen im einzelnen:
L schaltet den Literal-Suchmodus ein (s.u.).
T schaltet das Gegenteil, nämlich den Token-Suchmodus ein (s.u.). Gibt man sowohl L als auch T an (wie im Beispiel), dann gilt der letzte, also hier T. Wird aber keiner von beiden angegeben, dann wird automatisch der Token-Modus benutzt (Default).
K bewirkt, daß beim Suchen Klein- und Großbuchstaben unterschieden werden

(z.B. Apple <> APPLE). Läßt man das K weg, werden kleine und große Buchstaben beim Suchen gleich behandelt (z.B. ApPIE = aPpLe).
S bewirkt, daß die Zeile angezeigt und der Benutzer gefragt wird, ob tatsächlich ersetzt werden soll (S steht für „change some", wie bei EDASM und BIG MAC). Normalerweise antwortet man mit J oder N. Gibt man hingegen A für Abschalten ein, so wird es wie ein J gewertet und zudem die S-Option für die folgenden Ersetzungen wieder abgeschaltet.
V ist identisch mit der S-Option. Für Pascal-Benutzer: V(erify).
U unterdrückt die Anzeige von veränderten Zeilen.
Jeder andere Buchstabe führt zu einer Fehlermeldung.

2.2.3. Alt-, Neu-String und Delim

Die beiden Strings (Alt- und Neu-String) werden wie beim EDASM oder BIG MAC eingegeben, d.h. das erste Zeichen (außer Space), das weder zu den Option-Buchstaben noch zu den Zeilennummern gehört, wird als Start- und Endzeichen der Strings betrachtet, wobei das Endzeichen des Alt-Strings zugleich das Startzeichen des Neu-Strings ist. Dieses Zeichen nennt man Delimiter (Begrenzer), in der Syntaxzeile mit Delim abgekürzt. Der Delimiter muß ein Sonderzeichen sein (z.B. „'", „/", „!", „." usw.); Buchstaben oder Ziffern führen zu einer Fehlermeldung. Ebenfalls zulässig sind die Control-Zeichen ab Ctrl-D, sofern sie eingetippt werden können. Zu beachten ist, daß Komma und Bindestrich nur als Delimiter akzeptiert werden, wenn sie nicht die ersten Zeichen nach CHANGE sind. In diesem Fall würden sie nämlich zum Zeilenbereich gezählt werden.
Das *Anführungszeichen* (") hat noch eine Spezialfunktion: Wird es als Delimiter benutzt, dann werden die Strings *nicht* auf Applesoft-Schlüsselwörter untersucht und in Tokens kodiert.
Ebenfalls unkodiert bleiben natürlich alle Zeichen, die hinter einem REM oder DATA oder innerhalb von Anführungszeichen stehen, wie dies sonst auch der Fall ist. Einige Beispiele für die Strings und deren Delimiter:

CHANGE 1000-1399 "Hallo"Guten Tag"
Hier wird ab Zeile 1000 bis einschließlich Zeile 1399 jedes „Hallo" oder „HALLO" oder „hAllO" usw. durch „Guten Tag" ersetzt, und zwar auch eventuelle Variablen mit dem Namen HALLO.

CHANGE .PRINT D$.PRINT CHR$(4).
Da der Delimiter nicht das Anführungszeichen, sondern der Punkt ist, werden nur diejenigen PRINTs ersetzt, die als Applesoft-Tokens vorliegen. Eine Zeile wie 100 REM PRINT D$ startet DOS-Befehl würde nicht berührt werden, da hier der Befehl nicht als Token, sondern unkodiert in Form von 5 ASCII-Zeichen vorliegt.

CHANGE K.PRINT "Ich...".PRINT "Du...".
An diesem Beispiel sieht man, daß der Delimiter nur das Stringende markiert, wenn er nicht innerhalb von Anführungszeichen steht. Das Beispiel würde ferner wegen der K-Option weder „ICH" noch „ich" ersetzen, sondern nur „Ich".

2.3. Token und Literal

Im Token-Modus wird der Alt-String nur ersetzt, wenn er durch ein Applesoft-Schlüsselwort, Sonderzeichen oder Zeilenende abgegrenzt ist und mithin nicht, wenn er als Teil eines Wortes gefunden wurde. Beispiele:

CHANGE 10 .RAM.SPEICHER.
oder
CHANGE 10 T .RAM.SPEICHER.
angewendet auf die Zeile
10 RAM$ = "/RAM":
PRINT "Die RAM-Disk heißt";RAM$
ergibt
10 SPEICHER$ = "/SPEICHER":
PRINT "Die SPEICHER-Disk heißt"; SPEICHER$
(Jedes „RAM" wurde verändert.)
Eine Zeile wie
20 RAMDISK = 1:
PRINT "Das SUPER-PROGRAMM! Programmiert vom Ramdisk-Cracker"
würde unverändert bleiben, da „RAM" nie abgegrenzt vorkommt.
Im Literal-Modus, also mit
CHANGE L .RAM.SPEICHER.
entstünde daraus aber
20 SPEICHERDISK = 1:
PRINT "Das SUPER-PROGSPEICHERM! ProgSPEICHERmiert vom SPEICHERdisk-Cracker"
Alles klar?

2.4. Bildschirmausgabe

Wenn nicht die U-Option (= „Unterdrükkung der Ausgabe") benutzt wurde, wird jede Zeile, in der etwas ersetzt wurde, aufgelistet. Wird in der Zeile mehrfach etwas ersetzt, dann wird sie immer nur an der jeweiligen Stelle neu gelistet, so daß sie erst am Ende in der endgültigen Fassung sichtbar ist.
Weitere Ausgaben gibt es bei Fehlermeldungen oder der Abfrage, ob eine Ersetzung vorgenommen werden soll, falls die S-Option (= „Change Some") oder die V-Option (= „Verify") eingeschaltet wurde.
Damit man bei aktiver U-Option dennoch weiß, daß der Rechner nicht abgestürzt ist, ertönen unregelmäßige Knack- und Pfeiflaute aus dem Lautsprecher, solange der Rechner arbeitet.

2.5. Unterbrechung und Fehler

Der Such- und Ersetzvorgang kann jederzeit mit Ctrl-C unterbrochen werden, auch wenn der Cursor zu sehen ist. Es erscheint wie gewohnt ‚BREAK IN Zeilennummer'. Nach der Ausgabe von BREAK kann es bei langen Programmen noch kurze Zeit dauern, bis der Applesoft-Prompt wieder erscheint. Dies ist normal, denn die Zeilen-Linkadressen müssen noch neu berechnet werden.
Enthält die Eingabezeile vor den Strings ein Ctrl-C, so wird mit einem einfachen BREAK angehalten. Die Strings selbst können jedes eintippbare Ctrl-Zeichen enthalten.
Wurde ein Alt-String entdeckt, dann wird zunächst geprüft, ob die Zeile nach dem Ersetzen des Neu-Strings zu lang (> 250 Zeichen) werden würde. Ist das der Fall, erscheint die Zeilennummer und die Meldung „WIRD ZU LANG! E - ENDE, W - WEITER:". Antwortet man mit W, dann werden in dieser Zeile keine Ersetzungen mehr vorgenommen, aber in der nächsten Zeile wird normal weitergesucht. Mit E dagegen wird der Suchvorgang mit einem Applesoft-Warmstart abgebrochen (Das Programm bleibt natürlich intakt). Jedes andere Zeichen wird mit einem Beep abgelehnt.
Wenn das ganze Programm zu lang werden sollte (Programmende > HIMEM:), erscheint „PROGRAMM WIRD ZU LANG!".

Wichtiger Hinweis, auch im Hinblick auf Editoren: *Auf keinen Fall darf die Reset-Taste betätigt werden, während der Change-Befehl ausgeführt wird.* Am besten auch nicht mit Ctrl-C unterbrechen, da auch beim PRODOS.EDITOR und MACROEDITOR die Befehle Renumber usw. nicht mit Ctrl-C und Reset abgebrochen werden dürfen.

3. Der REMS-Befehl

3.1. Zweck

Die Art, wie Applesoft Programmzeilen auflistet, hat einen störenden Schönheitsfehler: Vor und nach jedem reservierten Wort wird ein Leerzeichen ausgegeben, also auch *nach* REM und DATA. Editiert man nun eine gelistete Zeile mit einer Escape-Sequenz und drückt <Return>, dann werden beim Kodieren dieser Zeile die neuen Leerzeichen wieder entfernt,

außer es handelt sich um eine REM- oder DATA-Zeile. Denn dann wird das Leerzeichen nach REM bzw. DATA in das Programm übernommen, und beim nächsten LIST erscheinen dann 2 Leerzeichen an dieser Stelle. Editiert man diese Zeile wieder, bekommt man 3 Leerzeichen, beim nächsten Mal 4 und so weiter.
(Da im PRODOS.EDITOR und MACROEDITOR die zu redigierende Zeile über die normale LIST-Routine in einen Zwischenpuffer eingelesen und dann am Bildschirm angezeigt wird, tritt auch hier der Space-Fehler auf. Man beachte auch den technischen Hinweis im Peeker, 6/1986, S. 43. us)
Ein kleine Routine, die alle Leerzeichen, die unmittelbar auf REM oder DATA folgen, auf einen Schlag entfernt („killt"), ist also recht nützlich und kann zur besseren Lesbarkeit des Programms beitragen. Diese kurze Routine wurde deshalb mit in die BASIC.SYSTEM-Erweiterung aufgenommen, da der PRODOS.EDITOR und der MACROEDITOR nur einen Befehl für das Entfernen *aller* REM-Zeilen oder *aller* Spaces *einer* REM-Zeile enthält.

3.2. Aufruf

Die Routine wird aufgerufen mit
REMS (steht für Remove Space)
Auf REMS folgende Zeichen werden ignoriert. Der Rems-Befehl kann immer dann benutzt werden, wenn der Change-Befehl auch zur Verfügung steht (s.o.).

3.3. Unterbrechung und Fehler

Wie beim Change-Befehl kann der Rems-Befehl im äußersten Notfall mit Ctrl-C abgebrochen werden. Die Zeit zwischen der BREAK-Meldung und dem Erscheinen des Applesoft-Prompts kann hier aber noch länger sein. Die Reset-Taste ist wie beim Change-Befehl genauso tödlich für das Applesoft-Programm, sobald ein Leerzeichen entfernt worden ist.
Da das Applesoft-Programm nur kürzer werden kann, kann kein „OUT OF MEMORY"-Fehler auftreten.

4. CHANGE.REMS für ProDOS

Sollten Sie ein reiner Applesoft-Programmierer sein, können Sie hier mit dem Lesen aufhören. Für Assembler-Programmierer aber soll das Programm wenigstens in groben Zügen erläutert werden. Details können dem Listing entnommen werden.
Das Programm CHANGE.REMS auf der Diskette besteht aus zwei Teilen:
1. dem Relokator und Mover sowie
2. dem eigentlichen Programm.
Beim Starten läßt sich der Relokator/Mover über die BASIC-SYSTEM-Routine ab $BEF5 sechs Speicherseiten (Pages) =1536 Bytes freimachen und schützen. Näheres hierzu kann man dem Kapitel „Implementierung eines externen Befehls" in „Apple ProDOS"; Bd. 2, S. 66ff. entnehmen.
Die Startadresse der sechs Pages wird von der Routine erst jetzt ermittelt und kann verschieden sein. Der eigentliche Code wird nun mit RELOCATE von $6000 in den neuen Speicherbereich kopiert und lauffähig gemacht. RELOCATE benutzt INSTDSP2 ab $F88C. Diese Routine stellt die Länge (1, 2 oder 3) des Befehls, auf den die Zero-Page-Stelle $003A-$003B zeigt, fest und speichert die Länge-1 (0, 1 oder 2) in $002F.
Der Second-Command-Handler-JMP-Befehl ab $BE06 wird auf DISPATCH gesetzt. Dann wird mit RTS zurückgekehrt. Wenn nun das BASIC.SYSTEM einen DOS-Befehl ausführen soll, also nach Return oder PRINT CHR$(4), aber keinen erkennt, wird zuerst $BE06 aufgerufen. Normalerweise zeigt $BE06 auf ein RTS, jetzt aber auf DISPATCH. Die I/O-Vektoren ($0036-$0039) zeigen dann jedoch nicht ins BASIC.SYSTEM, sondern auf die tatsächlichen I/O-Routinen. DISPATCH untersucht nun, ob Direktmodus vorliegt, und wenn ja, ob die Befehlszeile mit REMS oder CHANGE anfängt. Nur wenn beides nicht der Fall ist, wird RTS – mit gesetztem Carry – ausgeführt.
Wurde aber ein Befehl erkannt und soll nach Ausführung dieses Befehls über JMP direkt ins Applesoft zurückgekehrt werden, müssen die I/O-Vektoren mit $9A17 wieder auf das BASIC.SYSTEM gesetzt werden. (Achtung: $9A17 ist nicht offiziell!)

Change-Befehl

Der Change-Befehl untersucht nun die Parameter (ab DOCHANGE) und geht in die Change-Hauptschleife LINELOOP. LINELOOP wird immer angesprungen, wenn ein Zeilenende erreicht wurde. Wurde das Programm- oder Zeilenende überschritten, so wird die Routine beendet. Sonst wird der Abstand zum Zeilenende bestimmt, ein Zeiger ($00B8-$00B9) auf das erste Zeichen der Zeile gesetzt und in der Zeilenschleife CHECKNEXT fortgefahren. Ist der Abstand zum Zeilenende größer als der Alt-String lang ist, wird verglichen, ob die Zeichen, auf die der Zeiger verweist, mit dem Alt-String übereinstimmen. Ist das nicht der Fall, so wird der Zeiger um 1 erhöht, bei CHECKNEXT fortgefahren, dort der Abstand zum Zeilenende um 1 erniedrigt und weiterverglichen.
Wurde der Alt-String (unter Beachtung des Token-Modus) jedoch entdeckt, dann wird er ersetzt, wenn die Zeile nicht zu lang werden würde. Direkt vor dem Ersetzen wird mit MOVE der Rest des Programms verschoben, um Platz zu schaffen, wenn der Neu-String länger als der Alt-String ist, oder um überschüssigen Platz zu entfernen, wenn der Neu-String kürzer ist. Von nun an stimmen die Linkadressen der Applesoft-Programmzeilen nicht mehr! Deshalb müssen diese immer vor Verlassen des Programms neu berechnet werden, und zwar mit LINKSET ab $D4F2. Unglücklicherweise kehrt LINKSET nicht zurück, sondern geht direkt in den Applesoft-Warmstart (über JMP $D43C).
Nach dem Ersetzen wird der Zeiger auf das Zeichen hinter dem Neu-String gesetzt. Der Abstand zum Zeilenende wird aber um die Länge des Alt-Strings kleiner. Somit wird wieder bei CHECKNEXT fortgefahren.

Rems-Befehl

Nun zum Rems-Befehl: Er hat zwei Zeiger. Mit dem Zeiger „WOHER" wird das Programm ausgelesen, mit „WOHIN" wird es abgespeichert. Die beiden werden immer um dieselben Beträge erhöht, es sei denn, WOHER findet zu „killende" Leerzeichen. Dann wird nur WOHER erhöht, WOHIN aber nicht. Solange also kein Leerzeichen entfernt („gekillt") wurde, sind beide Zeiger gleich.
Wenn das Programmende erreicht wurde, wird WOHIN als neuer Programmendezeiger gesetzt und LINKSET angesprungen.
Wenn Ctrl-C gedrückt wird, erscheint die BREAK-Meldung ,und ein Flag wird gesetzt. Das Programm läuft aber weiter, nur werden keine Leerzeichen mehr übersprungen.

5. CHANGE.REMS.DOS.3.3

Die Peeker-Sammeldiskette enthält unter dem Namen CHANGE.REMS.DOS.3.3 den Objektcode der oben beschriebenen Routine, die separat oder in Verbindung mit dem MACROEDITOR verwendet werden kann. Im letzteren Fall ergibt sich folgende Speicherverteilung:
$7700-$7CFF: CHANGE.REMS
$7D00-$95FF: MACROEDITOR
HIMEM: liegt dann bei $7700, und die drei DOS-Puffer beginnen ab $9600. Die Befehlssyntax wurde dahingehend geändert, daß jetzt an Stelle der Befehlswörter CHANGE und REMS ein &-Befehl benutzt wird. Für das parameterlose REMS
heißt es jetzt
& DEL
und für CHANGE mit Parametern
& C

Danach erscheint „C" für Change, worauf man die Parameter in der gleichen Form wie bei dem ProDOS-Change-Befehl eingibt.

Kurzhinweise

1. Zweck:
Globale Änderung von Applesoft-Programmzeilen sowie Entfernung von Leertasten nach REM und DATA
2. Konfiguration:
Apple II+/e/c; ProDOS und DOS 3.3
3. Test:
BRUN CHANGE.REMS (ProDOS) oder BRUN CHANGE.REMS.DOS.3.3 (DOS).
4. Sammeldisk:
T.CHANGE.REMS
(Big-Mac-Quelltext)
CHANGE.REMS
(Objektcode für ProDOS; kann separat oder in Verbindung mit PRODOS.EDITOR verwendet werden)
CHAMGE.REMS.DOS.3.3
(Objektcode für DOS 3.3; Quelltext aus Platzgründen weder hier gelistet noch auf Sammeldisk befindlich; kann separat oder in Verbindung mit MACROEDITOR, vgl. Peeker, Heft 4/86, S.13, verwendet werden)

CHANGE.REMS

```
                  1   *
                  2   * CHANGE.REMS
                  3   * ============
                  4   *
                  5   PROGSTART EQU  $6000       Code ab $6000
                  6   *
                  7   * Erweiterung des Applesoft-Editors
                  8   * PRODOS.EDITOR und MACROEDITOR
                  9   * um die Befehle
                  10  * 1. CHANGE (Bereich)(Optionen).Alt$.Neu$
                  11  * 2. REMS
                  12  * von Frank Seide, Hamburg 1986
                  13  *
                  14  * Zero-Page-Variablen des Change-Befehls:
                  15  ALT       EQU  $201       Alt$ ab hier
                  16  NEU       EQU  $08        Adresse Neu$
                  17  ALTLEN    EQU  $1A        Länge Alt$
                  18  NEULEN    EQU  $1B        Länge Neu$
                  19  LENDIFF   EQU  $1C        Längendifferenz
                  20  DIFFVORZ  EQU  $1D        Differenzvorz.
                  21  LEN       EQU  $1E        Län.bearb.Zeile
                  22  RESTLEN   EQU  $D5        Abst.Zeilenende
                  23  DELIM     EQU  $FE        Delimiter
                  24  BLINE     EQU  $FA        Startzeilennr.
                  25  ELINE     EQU  $FC        Endzeilennr.
                  26  LAST      EQU  $D0        letztes Zeichen
                  27  SOME      EQU  $D1        Abfrage ?
                  28  TOKEN     EQU  $D2        Token-Search ?
                  29  AUCHKLEI  EQU  $D3        Klein vergl. ?
                  30  NOLIST    EQU  $D4        Ausgabe unterdr.
                  31  SCHONDA   EQU  $D7        Zeile auf Schirm
                  32  ZPTR      EQU  $DC        Zeilenanfg.
                  33  ZINV      EQU  $DA        INVERSE hier
                  34  ZNORM     EQU  $DB        NORMAL hier
                  35  LINCOUNT  EQU  $DF        Anz. Schirmzln.
                  36  *
                  37  * ZP-Variablen des Spacekillers:
                  38  WOHER     EQU  $B8        Auslesezeiger
                  39  WOHIN     EQU  $CE        Ablegezeiger
                  40  BRKFLAG   EQU  $D0        CtrC gegeben ?
                  41  *
                  42  * Oft benutzte Routinen:
                  43  CHRGET    EQU  $B1        nächstes Zeichen
                  44  CHRGOT    EQU  $B7        selbes Zeichen
                  45  LINKSET   EQU  $D4F2      Linkadressen
                  46  ADDY      EQU  $D998      $B8:=$B8+Y
                  47  ADDA      EQU  $D99A      $B8:=$B8+A+C
                  48  *
                  49            ORG  PROGSTART-118
                  50  *
                  51  ** Relokator & Mover **
                  52  *
                  53  * vom BASIC.SYSTEM Speicherplatz reservieren
                  54  * lassen und Startadresse merken:
5F8A: A9 06       55            LDA  #>PROGLEN+256 benötigte Pages
5F8C: 20 F5 BE    56            JSR  $BEF5      zuweisen lassen
5F8F: C9 66       57            CMP  #>PROGSTART+PROGLEN+256
5F91: B0 06       58            BCS  *+8        zu niedrig ?
5F93: 20 17 9A    59            JSR  $9A17      Ja: Reconnect
5F96: 4C 10 D4    60            JMP  $D410      Out of memory
5F99: 8D 00 60    61            STA  STPAGE     Start-Page merken
                  62  *
                  63  * Dort lauffähig machen (relozieren):
5F9C: E9 60       64            SBC  #>PROGSTART Adressdifferenz
5F9E: 85 1E       65            STA  LEN        merken
5FA0: A9 8A       66            LDA  #CODEBEGIN Codeadresse
5FA2: A0 60       67            LDY  #>CODEBEGIN
5FA4: 85 3A       68  RELOCATE STA  $3A         als PC
5FA6: 84 3B       69            STY  $3B
5FA8: 2C 30 C0    70            BIT  $C030      Ausstiegs-Knack
5FAB: C9 B6       71            CMP  #PROGSTART+PROGLEN fertig ?
5FAD: 98          72            TYA
5FAE: E9 65       73            SBC  #>PROGSTART+PROGLEN
5FB0: B0 1E       74            BCS  RELCATED   Ja=< RTS
5FB2: A2 00       75            LDX  #0         Befehlslänge
5FB4: 20 8C F8    76            JSR  $F88C      mit INSTDS2
5FB7: A4 2F       77            LDY  $2F        holen
5FB9: C0 02       78            CPY  #3-1       3-Byte-Befehl ?
5FBB: D0 0E       79            BNE  RELONEXT   Nein
5FBD: B1 3A       80            LDA  ($3A),Y    Ja: greift auf
5FBF: C9 60       81            CMP  #>PROGSTART Code zu ?
5FC1: 90 08       82            BCC  RELONEXT
5FC3: C9 66       83            CMP  #>PROGSTART+PROGLEN+256
5FC5: B0 04       84            BCS  RELONEXT   Nein
5FC7: 65 1E       85            ADC  LEN        Ja: relozieren
5FC9: 91 3A       86            STA  ($3A),Y    und eintragen
5FCB: 20 53 F9    87  RELONEXT JSR  $F953       nächsten Befehl
5FCE: D0 D4       88            BNE  RELOCATE   (immer)
                  89  *
                  90  * dorthin kopieren:
5FD0: A0 00       91  RELCATED LDY  #0          Kopieren
5FD2: A9 60       92            LDA  #>PROGSTART
5FD4: 84 3C       93            STY  $3C        von ($3C)
5FD6: 85 3D       94            STA  $3D
5FD8: AD 00 60    95            LDA  STPAGE
5FDB: 84 3E       96            STY  $3E        nach ($3E)
5FDD: 85 3F       97            STA  $3F
5FDF: A2 06       98            LDX  #>PROGLEN+256 Anzahl Pages
5FE1: B1 3C       99  COPYPROG LDA  ($3C),Y     Byte holen
5FE3: 91 3E       100           STA  ($3E),Y    kopieren
5FE5: C8          101           INY
5FE6: D0 F9       102           BNE  COPYPROG
5FE8: E6 3D       103           INC  $3D        (Pagegrenze)
5FEA: E6 3F       104           INC  $3F
5FEC: CA          105           DEX             alle Pages ?
5FED: D0 F2       106           BNE  COPYPROG   Nix is=< weiter
5FEF: A9 4C       107           LDA  #$4C       2nd Cmd Handler
5FF1: 8D 06 BE    108           STA  $BE06      des BASIC.SYSTEM
5FF4: AC 00 60    109           LDY  STPAGE
5FF7: A9 B0       110           LDA  #DISPATCH
5FF9: 8D 07 BE    111           STA  $BE07      setzen
5FFC: 8C 08 BE    112           STY  $BE08
5FFF: 60          113           RTS
                  114 *
                  115           DS   PROGSTART-*
                  116 *
                  117 ****** Datenbereich (nicht reloziert) ******
                  118 *
6000: 01          119 STPAGE   DFB  1           Startadresse Hi
                  120 *
6001: 53 4D 45    121 KILLWORT ASC  'SMER' ASCII für REMS
6004: 52
6005: 45 47 4E    122 CHGWORT  ASC  'EGNAHC' & CHANGE rückw.
6008: 41 48 43
                  123 *
600B: 20 57 49    124 LNZULANG ASC  ' WIRD ZU LANG!'
600E: 52 44 20 5A 55 20 4C 41
6016: 4E 47 21
6019: 20 45 20    125           ASC  ' E - ENDE, W - WEITER: '
601C: 2D 20 45 4E 44 45 2C 20
6024: 57 20 2D 20 57 45 49 54
602C: 45 52 3A 20
6030: 00          126           DFB  0
                  127 *
6031: 50 52 4F    128 PGZULANG ASC  'PROGRAMM WIRD ZU LANG! BREAK'
6034: 47 52 41 4D 4D 20 57 49
603C: 52 44 20 5A 55 20 4C 41
6044: 4E 47 21 20 42 52 45 41
```

peeker Börse

Biete Hardware

Apple IIc + Monitor Zenith + Maus+ 2. Laufwerk Distar, neuw. 2500,– Sa-So Tel. 0 82 93 / 14 78.

Apple IIe, 80Z+64K, Monitor, 2LW, Z80 EPSON RX80 FT, evtl. auch einzeln abzugeben, DM 3350,– Tel. 0 23 73/8 29 94

Verkaufe Apple IIC, + Monitor IIC + Maus, 2. Laufwerk, Epson RX-80F/T+, Appleworks umfangreiche Software + Literatur, NP ca. DM 6000,– für DM 3100,– Fa. Integral GmbH, Tel. 0 22 37/6 15 97.

IBS AP20 512 K, CP/M 68K, C-Comp. Editor, VHB DM 990,– V. Hatz, Neureuter Hauptstr. 229, 75KA 31.

Wir bieten Applied engineering: Ramworks 512K IIe 800,– Z-RAM 999,– Ramfactor 512K IIe/II+ 850,– Transwarp 3,6 × schneller 825,– Bitter Labortechnik 3050 Steinhude Tel. 0 50 33/15 22

Superpreis für APPLE II e/II + Zusatzkarten TEACH 80 Zeichenkarte DM 35,00 TEACH 16 K Speichererweiterung DM 35,00, TEACH PDI 1 parallel Drukkerinterface (listing) DM 80,00, TEACH PDI 3 parallel Druckerinterface (Grafik) DM 120,00, TEACH PDI 20 Druckerinterface Commodore DM 55,00. Alle Preise incl. Mehrwertsteuer zuzüglich Versandkosten. TEACH-Computer, Siemensstr. 22, 7257 Ditzingen, Telefon 0 71 56/3001-34

Apple IIc, Monitor, 2. LW, Mouse, Softw. + Lit., DM 2600,– Tel.: 0 23 61/49 14 30.

Apple IIe, DUODISK, Monitor Z80 80Z + 64K, Drucker eX80FT, Grafik-Interface, Joyst., Software VB: DM 4000,– Tel. 0 23 81/5 75 56.

Macintosh 128 auf 512KB DM 299,–. Fa. Schlösser, Tel. ab 17.00 Uhr, 089 - 98 58 89

Apple-IIe 128K 80Z CP/M Ramkarte SSC Karte 3 Drives 2 Controller Binder-Drucker- 8510 BNC-Monitor 2ext-Tastaturen IBM-Gehäuse div. Software+Literatur+Manuals DM 3500,– U. Merz Tel. 0 26 07/44 72.

Drucker OKI ML 80 DM 499,– ALS CPM+ Card: Z80B GMHz 64K Byte CPM 3.0 für Apple II DM 599,– Tel.: 0 71 42/6 51 08.

Original IIe mit 80Z 64K Z80 Karte Monitor 1 Laufwerk DM 1650,– Tel. 0 23 73/6 31 59.

512-KB-RAM auf 1-MB erweiterbar + RD-Treiber (Pascal, ProDOS, CP/M, DOS) DM 750,– Klaus Seiler, W.- Andreas-Allee 1, KA (07 21/2 97 78)

1 MByte Ramkarte incl. Softw. DOS ProDOS, UCSD, CPM VHB DM 800,– Stibane Hardenberg 34, 1 Berlin 12

Farbmonitor Commodore 1701 (auch für andere Systeme mit Videoausgang geeignet. 40Z/ Zeile: DM 400,– Tel. 06 31/2 28 94.

Schweiz: Apple IIe-komp. mit Z80, 2.LW, Grappler, ser. Interface, Joystick, Eprommer, Monitor, viel Doku, viel SW, VB sFr. 2000,– T. Felner Quellenstr. 30, 8005 Zürich, Tel. 01/44 17 71.

IBS AP22 (Z80H, 8MHz, 64K), Buch, Patch, VB DM 450,- Tel.: 0511/211597 oder 0511/ 211159.

Frei prog. Tastatur für II plus (ähnl. Brose siehe UM2/85) für DM 350,– (DM 695,–) Tel.: (0 62 21) 2 62 21/2 02 50 Kehrel.

RAMWORKS II mit 1MB RAM, original VB sFr. 600,– AK. Koppler dataphon s21d, VB sFr. 200,–, 0041-1-44 1771.

Apple II + System VB 550, M. Horak, Baumeisterweg 12, 7000 Stuttgart 1.

Apple IIe 128KB Monitor II Duodisk 80Z Z80 SSC Imagewriter Mouse DM 3800,– T.05085/1367.

Suche Hardware

Koala Pad Touch Tablet incl. SW D. Brünn, 040/5 23 28 39 ab 18.00 Uhr.

Biete Software

Applesoft-Module: TOOLBOX, Stringbox, Window-Box, AMPER-Box jeweils nur DM 35,– von Ernst Heinz, Püttkampsweg 13, 2000 Hamburg 52.

Public Domain Software für Apple Liste DM 10. Ueding electronics Holtewiese 2, 5750 Menden 1 Tel. 0 23 73/6 31 59.

Software Uhr für Apple II+, e, c, Zeitschaltmöglichkeit Diskette + Anleitung DM 25,– Oecking Tel: Do. 02 31 / 39 19 20

Pascal-Module: TOOLBOX DM 50,– REAL-Erweiterung DM 25,–, LORES-Grafikbox DM 30,– von Ernst Heinz, Püttkampsweg 13, 2000 Hamburg 52.

Topsoftware für Apple II: Fußballtabellenverw. DM 30,–, Gamemaker: 150,–, Fortrun: 550,–. Info (0,80 DM) bei: W. Rittmeyer, Wehrbruchweg 30, 4060 Viersen 1.

Apple II: DFÜ-Kermit, Pascal satt Public Domain in DOS u. CP/ M.Je Volume DM 15,–. Bahnhofssimulation, Sprachen (S.A.L.), Schulprogr. Gratisinfo: Fa. Waltraud Muhle, Waldwinkel 3, 2105 Seevetal 3.

––– STOCKMASTER II ––– Das Apple II-Programm für echte Börsengewinne bei: TÖNGI Computerpraxis, In der Hofreite 11, D-6200 Wiesbaden.

Library für Kyan Pascal-2.0: mit diversen Routinen: nur DM 10,– (Schein/Scheck). Von R. Krause, Finkenweg 4, 4902 Bad Salzuflen 5.

Verkaufe für Apple IIe/ CP/M 2.2 C-Compiler + MicroShell Preis VH Tel. 0228/732554/238929.

Suche Software

Schreiner Programme für Apple IIe Pascal CPM DOS oder PRODOS sucht Heinz Heubrock, Osterbauer 101, 4715 Ascheberg.

Suche für Apple IIc gebr. Software für kommerz. Anwendung Tel. 074/74 20 01.

Suche Software für Apple IIe,c! Tel. 0 92 51 / 54 76.

Kontakte

Apple-AG gegründet!!! Ab sofort gibt es eine bundesweite DE-HOCA Apple-User-Groupe mit Programmservice, Kontakt: J. Reinhardt, Minden, Tel.: 05 71/ 3 65 71.

Tausch

Tausche Apple comp. 64K, Z80, CPU, IBM-Gehäuse, separate Tastatur 80-Zeichen, 2 Laufwerke 256K und Monitor gegen Apple IIc + Monitor. Tel. 0 24 06 / 7 98 83.

Verschiedenes

APPLE REPARATUREN
(auch compatible M-boards, z.B. **Atlas, Arca, CES, Datastar, Dipa, Lasar, Mewa, PC-48 + 64, Plato, Radix, o. ae.) sowie Zusatzkarten und Disk-Drives** führt unser Spezialistenteam mit mehr als 5-jähriger Kunden- und Reparatur-Dienst-Erfahrung, garantiert zuverlässig und besonders kostengünstig aus.
Bitte genaue Fehlerangabe sowie Tel. Nr. für evtl. Rückfragen nicht vergessen.
Auf Wunsch Kostenvoranschlag.
aaa-electronic gmbh
Habsburgerstr. 134, 7800 Freiburg, Tel. 0761/276864, Tx. 772642aaad

Endlich–Wordstar an Star SG10 ob NLQ od. Engschrift. Wir patchen Ihren WS für den Drucker SG10. für DM 40,– Bitte rufen Sie uns an. n.19.00h Tel. 09661/1300. Biesler 8458 Sulzbach Max-Planck-Str. 12.

Disketten
*5 1/4" **ab DM 1,20 /** DM 1,75*
3 1/2", 135 tpi, DM 4,10/5,60
auch andere, 6 Mon. Garant.
Allg. Austro-Ag, Ringstr. 10
D-8057 Eching, Tel. 08133/6116

Medizin. Doktorarb. zu vergeben ges. med. Stud. mit guten Kenntniss. in Apple-Grafik + Assembler für Rekonstr.-Progr. in histologie. Zuschr. an Dr. Lechleuthner Anat. Anst. Pettenkoferstr. 11, 8000 München 2

Ramworks IIe (Batterie gepuff.) Accelerator, Uhr Z80. 80Z zu verk. Tel. Mo-Fr. 20-21 Uhr 02 12/5 19 76.

Apple IIc, 2 Laufwerke, Z80-Karte, Joy (Orig) + Software (100 Disk)+ Bücher + Zeitsch. nur DM 2900,– Tel.: 069/7 89 42 67.

Wer hilft? Welche Hardware-Änderungen muß ich machen an meinem Diskdriver Remex RFDG60 damit er einwandfrei mit Apple II + Ehring Controller arbeitet (Holland) P. Visser Marykestr. 23 Noordwyk.

!!!! Da gibt's was umsonst !!!! 4 × im Jahr den neuen Katalog. Bühler Elektronik, Postfach 32, 7570 Baden-Baden

Ihre Erphi-Vertretung für die Schweiz: Beltronic, Im Chapf, 8455 Rüdlingen, Tel. 0 18 67 31 41, Telex 8 25 981.

Für Ihre Unterlagen

Abonnement bestellt

am: _____

Vertrauensgarantie:
Ich habe davon Kenntnis genommen, daß ich die Bestellung schriftlich durch Mitteilung an den Dr. Alfred Hüthig Verlag GmbH, Postfach 10 28 69, 6900 Heidelberg innerhalb von 7 Tagen widerrufen kann. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs (Datum des Poststempels).

Peeker
Leserservice
Postfach 10 28 69
6900 Heidelberg

Für Ihre Unterlagen

Folgende Bücher bestellt:

_____

_____

_____

_____

_____

am: _____

bei:

Peeker
Versandbuchhandlung
Postfach 10 28 69
6900 Heidelberg 1

Für Ihre Unterlagen

Folgende Disketten
und Programme bestellt:

_____

_____

_____

_____

_____

am: _____

bei:

Peeker
Softwareabteilung
Postfach 10 28 69
6900 Heidelberg 1

PEEKER

Abo-Karte

Ja, ich möchte **Peeker** abonnieren.
Liefern Sie mir **Peeker** ab Ausgabe ..... zum Jahresbezugspreis von z. Zt. DM 72,– (Inland) inkl. MwSt. Die Lieferung erfolgt frei Haus. Porto, Verpackung und Zustellgebühren übernimmt der Verlag. Der Jahresbezugspreis für das Ausland beträgt z. Zt. DM 72,– plus DM 18,– Versandspesen.

X

Datum _____ 1. Unterschrift _____

Bitte lesen!

Vertrauensgarantie: Ich habe davon Kenntnis genommen, daß ich die Bestellung schriftlich durch Mitteilung an den Dr. Alfred Hüthig Verlag GmbH, Postfach 10 28 69, 6900 Heidelberg innerhalb von 7 Tagen widerrufen kann. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs (Datum des Poststempels).

X

Datum _____ 2. Unterschrift _____

Verlagshinweis: Das Abonnement verlängert sich zu den jeweils gültigen Bedingungen um ein Jahr, wenn es nicht 2 Monate vor Jahresende schriftlich gekündigt wird.

Wir können nur Bestellungen mit zwei Unterschriften bearbeiten.

PEEKER

Buch-Karte

Bitte senden Sie mir gegen Rechnung folgende Bücher:

- ☐ Bühler, Applesoft-BASIC, 3-7785-1094-0, DM 38,–
- ☐ Eggerich, dBase II, Bd. 1, 3-7785-1147-5, DM 39,80
- ☐ Eggerich, dBase II, Bd. 2, 3-7785-0987-X, DM 39,80
- ☐ Eggerich, dBase II, Bd. 3, 3-7785-0988-8, DM 39,80
- ☐ Gabriel, Applewriter, 3-7785-1234-X, DM 35,–
- ☐ Hagenmüller, Microsoft-BASIC, Bd. 1, 3-7785-1038-X, DM 38,–
- ☐ Juhnke/Redlin, Apple Pascal, Bd. 1, 3-7785-1246-3, ca. DM 40,–
- ☐ Kehrel, Apple Assembler lernen, Bd. 1, 3-7785-1151-3, DM 38,–
- ☐ Kehrel, Apple Assembler lernen, Bd. 2, 3-7785-1170-X, DM 38,–
- ☐ Schäpers, ProDOS Analyse, 3-7785-1134-3, DM 68,–
- ☐ Schäpers, Bewegte Apple-Graphik, 3-7785-1150-5, DM 58,–
- ☐ Stiehl, Apple DOS 3.3, 3-7785-1297-8, DM 28,–
- ☐ Stiehl, Apple ProDOS, Bd. 1, 3-7785-1098-3, DM 28,–
- ☐ Stiehl, Apple ProDOS, Bd. 2, 3-7785-1036-3, DM 30,–
- ☐ Stiehl, Apple Assembler, 3-7785-1047-9, DM 34,–
- ☐ Wassermann, Apple IIc Handbuch, 3-7785-1157-2, DM 35,–

Datum _____ Unterschrift _____

PEEKER

Software-Karte

Bitte senden Sie mir gegen Rechnung folgende Disketten:

- ☐ Peeker-Sammeldiskette, einzeln
  Disk# _____, Disk# _____
  Disk# _____, Disk# _____
  Preis je Disk DM 28,– (einzeln)
- ☐ Peeker-Sammeldiskette, im Fortsetzungsbezug ab Disk# _____ (Mindestbezug 6 Disketten) Preis je Disk DM 20,–
- ☐ Apple DOS 3.3, Begleitdisk., DM 28,–
- ☐ ProDOS, Band 1, Begleitdisk., DM 28,–
- ☐ ProDOS, Band 2, Begleitdisk., DM 28,–
- ☐ Apple Assembler, Begleitdisk., DM 28,–
- ☐ ProDOS-Editor 1.0, Programm, DM 98,–
- ☐ MMU 2.0, Programm, DM 98,–
- ☐ INPUT 2.0, Programm, DM 98,–
- ☐ Softbreaker 1.0, Programm, DM 20,–
- ☐ DB-Meister, Programm, DM 290,–
- ☐ Superquick, Programm, DM 48,–
- ☐ Turtle Graphics, Programm, DM 98,–
- ☐ Disk 40, Programm, DM 48,–
- ☐ Kyan-Pascal 2.0, Programm, DM 170,–
- ☐ Fast-Writer, DOS 3.3, DM 128,–
- ☐ Fast-Writer, ProDOS, DM 128,–
- ☐ Double-Hires-Tools für Applesoft, DM 28,–
- ☐ Double-Hires-Tools für Kyan, DM 28,–
- ☐ Kyan-Toolkit Nr. _____, DM _____

Datum _____ Unterschrift _____

Abo-Karte

Name

Firma

Straße

PLZ/Ort

Ich wünsche jährliche Berechnung durch:
☐ Verlagsrechnung ☐ Abbuchung von meinem Bank- bzw. Postscheckkonto

Bank/PschA

Bankleitzahl

Kto.-Nr.

POSTKARTE

Peeker
Leserservice

Dr. Alfred Hüthig Verlag GmbH

Postfach 10 28 69

6900 Heidelberg

INPUT 2.0

Ein Bildschirm-Maskengenerator für DOS 3.3 und ProDOS

von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7785-1021-5

„Input 2.0" liegt wahlweise in der Bank 1 oder Bank 2 der Language Card und wird durch einen kurzen Driver in den unteren 48K aufgerufen.
Für jedes Feld der Bildschirmmaske lassen sich u. a. definieren: Feldlänge (bis zu 255 Zeichen) – Vtab – Htab – Datentyp (insgesamt 8 Typen) – Scrollflag (starre oder dynamische Maske) – Ctrlflag – Füllflag – Löschflag – Bildschirmflag (40- oder 80-Z-Darstellung). Innerhalb eines Eingabefeldes besteht jeder denkbare Redigierkomfort (Insert, Delete, Rubout, Restore usw.).
Gerätevoraussetzung: Apple IIe oder IIc; ferner Apple II+ im 40-Zeichenmodus

Buch-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

POSTKARTE

Peeker
Buchabteilung

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

MMU 2.0

Memory Managements Utilities

für die Apple IIe 64K-Karte DOS 3.3 (und ProDOS)

von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7787-1023-1

Insgesamt enthält die neue „MMU 2.0"-Diskette über 25 Programme, die neue Einsatzmöglichkeiten für die Extended 80 Column Card (erweiterte 80-Z-Karte = 64K-Karte für den Apple IIe) erschließen. Ein Teil der Programme laufen auch auf dem Apple II Plus, doch ist „MMU 2.0" primär für 64K-Karte-Besitzer gedacht.

Gerätevoraussetzung: Apple IIe mit 64K-Karte oder IIc

Software-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

POSTKARTE

Peeker
Softwareabteilung

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

DISK 40

Disketten-Organisationsprogramm für DOS-3.3 - 35-40 Spuren

von Hermann Seibold und Dipl.-Ing. Udo Marin, 1986, Programmdiskette mit Anleitung, DM 48,–

Durch eine einfach zu bedienende Menüführung können DOS-3.3-Disketten umfangreich bearbeitet oder kopiert werden.
– Tabellarische Ausgabe der Diskettenbelegung
– Ordnen des Catalogs
– „Undelete"n von versehentlich gelöschten Dateien
– Vergleichen von Disketten, Dateien oder DOS-Spuren
– Kopieren von Disketten, Dateien oder DOS-Spuren
– Formatieren von Daten-Disketten
– Erweitern auf 40 Spuren bei bestehenden 35-Spur-Disketten
– Ändern des Boot-Programms
– File-Editor zum Editieren von Disketten-Dateien
– Komfortabler Sektor-Editor für Hex- und ASCII-Darstellung
–VTOC-Editor, z. B. zur Freigabe der DOS-Spuren

Hüthig Software Service, Postfach 10 28 69, D-6900 Heidelberg

Peeker-Sammeldisk #22

DOS-3.3-Diskette; Heft 8/1986 (restliche Programme) und 9/1986; Einzelpreis DM 28,–; Fortsetzungspreis DM 20,–; Hinweis: Das Pascal-Programmpaket PICEDIT von Heft 8/1986 befindet sich auf Sammeldisk #20

(1) = Zweck; (2) = Heft/Seitenzahl; (3) = Gerätekonfiguration; (4) = Betriebssystem; (5) = Programmstart; (6) = Sonstiges

A 009 SPURJUSTAGE.SIMULATION
A 003 MOTORTEST
(1) Laufwerk-Spurjustage; (2) Heft 10/86, S. 6; (3) II+/e; (4) DOS 3.3; (5) RUN SPURJUSTAGE.SIMULATION bzw. MOTORTEST; (6) Für Simulationsprogramm ist kein Meßgerät erforderlich.

T 084 T.CHANGE.REMS
B 008 CHANGE.REMS
T 089 T.CHANGE.REMS.DOS.3.3
B 008 CHANGE.REMS.DOS.3.3
(1) Change-Befehl für Applesoft-Programmierer; (2) Heft 10/86, S. 22; (3) II+/e/c; (4) DOS 3.3 für CHANGE.REMS.DOS.3.3, ProDOS für CHANGE.REMS; (5) BRUN CHANGE.REMS bzw. CHANGE.REMS.DOS.3.3; (6) siehe auch Heft 10/86, S. 15 wegen Macroeditor.

A 009 HEX.MONITOR
(1) Programm zur Eingabe von Hex-Dumps; (2) Heft 10/86, S. 36; (3) II+/e/c; (4) DOS 3.3; (5) RUN HEX.MONITOR

A 004 DHRCG.LOADER
T 043 T.DHRCG
B 006 DHRCG
B 005 GERMAN.SET
A 003 AMPER.DOUBLE.HIRES.BAS
B 004 AMPER.DOUBLE.HIRES
(1) 80-Z/Z-Bildschirmtreiber für Double-Hires-Schrift; (2) Heft 10/86, S. 38; (3) IIc oder IIe mit 64K-Karte; (4) DOS 3.3; (5) RUN DHRCG.LOADER

T 062 DISK1.TEXT
T 078 DISK2.TEXT
T 079 DISK3.TEXT
(1) Diskette neditor für UCSD-Pascal; (2) Heft 10/86, S. 44; (3) II+/e/c; (4) Apple-Pascal 1.1, 1.2; (5) E(ecute DISK; (6) Die Textfiles müssen zunächst mit GETDOS (von Disk #18) auf Ihre UCSD-Diskette konvertiert und dann zu DISK compiliert werden. Zum Compilieren nur DISK1.TEXT eingeben.

T 009 T.PRODOS.PATCH.111
B 003 PRODOS.PATCH.111
(1) ProDOS-Patch für Kompatible; (2) Quellcode-Nachtrag zu Heft 5/86, S. 60; (3) II+/e-Kompatible; (4) Nur für ProDOS 1.1.1; (5) Vor dem Programmstart Hinweise in 5/86, S. 60 beachten!

Hüthig Software Service · Postfach 102869 · 6900 Heidelberg

Peeker-Sammeldisk # 21

DOS-3.3-Diskette; Heft 8/1986 (restliche Programme) und 9/1986; Einzelpreis DM 28,–; Fortsetzungspreis DM 20,–; Hinweis: Das Pascal-Programmpaket PICEDIT von Heft 8/1986 befindet sich auf Sammeldisk #20

(1) = Zweck; (2) = Heft/Seitenzahl; (3) = Gerätekonfiguration; (4) = Betriebssystem; (5) = Programmstart; (6) = Sonstiges

Heft 8/1986

A 005 LOGIK
T 004 MLOGIK
(1) Übungsprogramme zur Aussagenlogik; (2) Heft 7/86, S. 32; (3) II+/e/c; (4) für LOGIK DOS 3.3 oder ProDOS, für MLOGIK CP/M mit MBASIC; (5) RUN LOGIK; (6) MLOGIK muß zunächst mit APDOS auf MBASIC-Diskette konvertiert werden.

T 019 T.NETZWERK
B 003 NETZWERK
(1) Vernetzung von Apple-Rechnern; (2) Heft 7/86, S. 50; (3) II+/e; (4) DOS 3.3; (5) BRUN NETZWERK; (6) nur mit entsprechender Hardware starten, siehe Aufsatz!

T 006 T.UNIFORMAT
B 002 UNIFORMAT
(1) Formatierbefehl für Unidisk; (2) Heft 7/86, S. 56; (3) II+/e mit Unidisk; (4) ProDOS; (5) BLOAD UNIFORMAT: CALL 768; (6) Slot beachten! Siehe Aufsatz.

Heft 9/1986

A 003 HARDBREAKER
A 006 GEN.EXEC
T 009 T.NMI
B 003 NMI
T 007 T.LCMOVE
B 002 LCMOVE
T 003 HARDBREAKER.INIT
T 003 HARDBREAKER.INIT.BASIS
T 002 SAVEMEM
(1) NMI-Interrupt-Routinen für entsprechende Experimentierplatine; (2) Heft 9/86, S. 34; (3) II+/e, evtl. auch Basis; (4) DOS 3.3; (5) RUN HARDBREAKER; (6) zur Platine siehe Aufsatz.

T 052 FILER.PAS
(1) Turbo-Pascal-Dateikopierprogramm; (2) Heft 9/86, S. 40; (3) II+/e/c mit CP/M-Karte; (4) Turbo-Pascal; (5) >FILER; (6) Programm zunächst mit APDOS auf Turbo-Disk konvertieren und dann compilieren.

T 026 DREIECK.TEXT
(1) Dreiecksberechnung in UCSD-Pascal; (2) Heft 9/86, S. 48; (3) II+/e/c; (4) UCSD-Pascal 1.1 oder 1.2; (5) Exec DREIECK; (6) Programm zunächst mit GETPAS auf UCSD-Disk konvertieren und dann compilieren.

T 024 STRINGUTILS.I
T 013 STRINGDEMOS.P
(1) Kyan-Pascal-String-Include-File mit Demo; (2) Heft 9/86, S. 50; (3) II+/e/c; (4) ProDOS und Kyan-Pascal; (5) %STRINGDEMOS; (6) Programme müssen erst mit CONVERT oder DOSTOPRO auf ProDOS-Disk konvertiert werden, dann STRINGDEMOS.P compilieren.

A 002 EINZEL.TASC
B 021 EINZEL.RUNTIME
B 061 EINZEL.OBJ
A 056 EINZEL.KALK
T 002 EINZEL.KONST
T 006 EINZEL.KLEIN
T 006 EINZEL.GROSS
(1) Bucheinzelkalkulationsprogramm für Verlage; (2) Heft 9/86, S. 56; (3) II+/e/c; (4) DOS 3.3; (5) RUN EINZEL.TASC oder RUN EINZEL.KALK

A 003 FUNKTIONEN START
A 011 FUNKTIONEN ANLEITUNG
B 002 FUNKTIONEN EXC
A 057 FUNKTIONEN
(1) Funktionsplotter für Apple II+; Gegenstück zu Funktionsplotter für IIe/c von Sammeldisk #18; (2) Heft 9/86, S. 62; (3) II+; (4) DOS 3.3; (5) RUN FUNKTIONEN START

Hüthig Software Service · Postfach 102869 · 6900 Heidelberg

```
6Ø4C: 4B
6Ø4D: Ø7 ØØ      129            DFB  7,Ø
                 13Ø  *
6Ø4F: 45 52 53   131  VFYTEXT   ASC  'ERSETZEN ? J/N, E - ENDE, A -
6Ø52: 45 54 5A 45 4E 2Ø 3F 2Ø
6Ø5A: 4A 2F 4E 2C 2Ø 45 2Ø 2D
6Ø62: 2Ø 45 4E 44 45 2C 2Ø 41
6Ø6A: 2Ø 2D 2Ø 41 4C 4C 45 3A
6Ø72: 2Ø
6Ø73: ØØ         132            DFB  Ø
                 133  *
6Ø74: 3F 56 45   134  INVDELIM ASC  '?VERBOTENER DELIMITER'
6Ø77: 52 42 4F 54 45 4E 45 52
6Ø7F: 2Ø 44 45 4C 49 4D 49 54
6Ø87: 45 52
6Ø89: ØØ         135            DFB  Ø
                 136  *
                 137  ****** ab hier relozierter Code ******
                 138  *
                 139  CODEBEGIN EQU  *          Code-Anfang
                 14Ø  *
                 141  *** Routinen für BASIC.SYSTEM-Einbindung ***
                 142  *
                 143  * beginnt Eingabezeile mit Befehlswort ? (-<C
6Ø8A: 86 4E      144  CHCKCMD  STX  $4E        Adresse Lo
6Ø8C: AD ØØ 6Ø   145           LDA  STPAGE     und Hi des
6Ø8F: 85 4F      146           STA  $4F        Befehlswortes
6Ø91: A2 FF      147           LDX  #-1
6Ø93: 2Ø A7 6Ø   148  CHCKCMDØ JSR  GETCHAR    Zeichen holen
6Ø96: 29 5F      149           AND  #$5F       Klein --< Groß
6Ø98: D1 4E      15Ø           CMP  ($4E),Y    stimmt überein ?
6Ø9A: DØ 33      151           BNE  NOCMD      Nein: SEC & RTS
6Ø9C: 88         152           DEY  Doch:      nächstes
6Ø9D: 1Ø F4      153           BPL  CHCKCMDØ   Befehlszeichen
6Ø9F: 2Ø 8B FD   154           JSR  $FD8B      alle: ClEoln&CR
6ØA2: 2Ø A7 6Ø   155           JSR  GETCHAR    nächstes Zeichen
6ØA5: 18         156           CLC  "Befehl stimmt"
6ØA6: 6Ø         157           RTS
                 158  *
                 159  * Ein Zeichen für Befehlsanalyse holen:
6ØA7: E8         16Ø  GETCHAR  INX  Zeiger     erhöhen
6ØA8: BD ØØ Ø2   161           LDA  $2ØØ,X     Zeichen holen
6ØAB: C9 AØ      162           CMP  #32+128    Space ?
6ØAD: FØ F8      163           BEQ  GETCHAR    Ja=< ignorieren
6ØAF: 6Ø         164           RTS
                 165  *
                 166  ** 2. Command-Handler (wird vom          **
                 167  ** BASIC.SYSTEM via JSR $BEØ6 aufgerufen) **
6ØBØ: A6 76      168  DISPATCH LDX  $76        Direktmodus ?
6ØB2: E8         169           INX  (Applesoft)
6ØB3: FØ Ø5      17Ø           BEQ  DISPATCHØ  Ja
6ØB5: AD 42 BE   171           LDA  $BE42      (BASIC.SYSTEM)
6ØB8: DØ 15      172           BNE  NOCMD      Nein: nichts erkennen
6ØBA: A2 Ø1      173  DISPATCHØ LDX  #KILLWORT  Zeiger auf KILL
6ØBC: AØ Ø3      174           LDY  #4-1       4 Zeichen lang
6ØBE: 2Ø 8A 6Ø   175           JSR  CHCKCMD    KILL-Befehl ?
6ØC1: BØ Ø3      176           BCS  *+5        Nein
6ØC3: 4C F6 64   177           JMP  KILLSPACE  Ja=< Spacekiller
6ØC6: A2 Ø5      178           LDX  #CHGWORT   Zgr. auf CHANGE
6ØC8: AØ Ø5      179           LDY  #6-1       6 Zeichen lang
6ØCA: 2Ø 8A 6Ø   18Ø           JSR  CHCKCMD
6ØCD: 9Ø Ø2      181           BCC  WASCHANGE  Ja
6ØCF: 38         182  NOCMD    SEC  Nein:      "kein
6ØDØ: 6Ø         183           RTS  Command    erkannt"
                 184  *
                 185  *** CHANGE-Programm ***
                 186  *
                 187  * Zeile begann mit CHANGE:
6ØD1: C9 8D      188  WASCHANGE CMP  #13+128    CHANGE >Ret.< ?
6ØD3: FØ 17      189           BEQ  CHANGEIN   Ja=< Zeile holen
6ØD5: AØ FF      19Ø           LDY  #-1        Nein: Zeile
6ØD7: C8         191  WASCHGØ  INY  benutzen,  dazu
6ØD8: B9 ØØ Ø2   192           LDA  $2ØØ,Y     alle Hi-Bits der
6ØDB: 29 7F      193           AND  #$7F       Eingabezeile
6ØDD: 99 ØØ Ø2   194           STA  $2ØØ,Y     löschen
6ØEØ: 49 ØD      195           EOR  #13        CR erreicht ?
6ØE2: DØ F3      196           BNE  WASCHGØ    Nein=< nächstes
6ØE4: 99 ØØ Ø2   197           STA  $2ØØ,Y     durch Ø ersetzen
6ØE7: CA         198           DEX  X/Y        auf
6ØE8: AØ Ø2      199           LDY  #2         Parameter setzen
6ØEA: DØ Ø5      2ØØ           BNE  DOCHANGE   & sie auswerten
                 2Ø1  *
                 2Ø2  * Nur CHANGE <Return> =< Parameterzeile holen:
6ØEC: A2 C3      2Ø3  CHANGEIN LDX  #'C'+128   'C' als Prompt
6ØEE: 2Ø 2E D5   2Ø4           JSR  $D52E      Zeileneingabe
                 2Ø5  *
                 2Ø6  * X/Y zeigt direkt vor 1. Parameterzeichen:
```

```
6ØF1: 86 B8      2Ø7  DOCHANGE STX  $B8        Zeiger merken
6ØF3: 84 B9      2Ø8           STY  $B9
6ØF5: 2Ø 17 9A   2Ø9           JSR  $9A17     BASIC.SYSTEM an!
                 21Ø  *
                 211  ** Parameterzeile via CHRGET auswerten **
                 212  * Zeilenbereich nach BLINE/ELINE:
6ØF8: A2 ØØ      213           LDX  #Ø         Defaultwerte:
6ØFA: 86 FA      214           STX  BLINE      Startzeile=Ø
6ØFC: 86 FB      215           STX  BLINE+1
6ØFE: CA         216           DEX  =<         X=$FF
6ØFF: 86 FC      217           STX  ELINE      Endzeile=65535
61Ø1: 86 FD      218           STX  ELINE+1    (=$FFFF)
61Ø3: 2Ø B1 ØØ   219           JSR  CHRGET     folgt Ziffer ?
61Ø6: BØ ØB      22Ø           BCS  NOBLINE    Nein
61Ø8: 2Ø Ø5 63   221           JSR  INNUM      Ja: Wert holen
61ØB: 86 FA      222           STX  BLINE       als Startzeile
61ØD: 84 FB      223           STY  BLINE+1
61ØF: 86 FC      224           STX  ELINE       und Endzeile
6111: 84 FD      225           STY  ELINE+1
6113: C9 2D      226  NOBLINE  CMP  #'-'       folgt '-'
6115: FØ Ø4      227           BEQ  *+6
6117: C9 2C      228           CMP  #','       oder ',' ?
6119: DØ 12      229           BNE  NOELINE    Nein
611B: A2 FF      23Ø           LDX  #$FF       Ja=< Endzeile
611D: 86 FC      231           STX  ELINE      ist 65535
611F: 86 FD      232           STX  ELINE+1    (=$FFFF)
6121: 2Ø B1 ØØ   233           JSR  CHRGET     folgt Ziffer ?
6124: BØ Ø7      234           BCS  NOELINE    Nein
6126: 2Ø Ø5 63   235           JSR  INNUM      Ja: Wert holen
6129: 86 FC      236           STX  ELINE      als Endzeile
612B: 84 FD      237           STY  ELINE+1
                 238  *
                 239  * Optionen auswerten:
612D: 46 D1      24Ø  NOELINE  LSR  SOME       Defaultwerte:
612F: 38         241           SEC  "Abfrage": Nein
613Ø: 66 D2      242           ROR  TOKEN      "Token": Ja
6132: 46 D3      243           LSR  AUCHKLEI   "Klein": Nein
6134: 46 D4      244           LSR  NOLIST     "Listen": Ja
6136: 2Ø B7 ØØ   245           JSR  CHRGOT     Optionszeichen
6139: 2Ø 1Ø 63   246           JSR  MODEINPUT  auswerten
613C: 46 D3      247           LSR  AUCHKLEI   Bit 7 -< Bit 6
                 248  *
                 249  * Strings kodieren & auswerten:
613E: 2Ø 77 63   25Ø           JSR  GETSTRGS
                 251  *
                 252  * Startadresse der Startzeile holen:
6141: A5 FA      253           LDA  BLINE      Zeilennummer
6143: A4 FB      254           LDY  BLINE+1    nach
6145: 85 5Ø      255           STA  $5Ø        LINNUM
6147: 84 51      256           STY  $51
6149: 2Ø 1A D6   257           JSR  $D61A      FNDLIN aufrufen
614C: A5 9B      258           LDA  $9B        Adresse steht
614E: A4 9C      259           LDY  $9C        in LOWTR. Nun
615Ø: 85 B8      26Ø           STA  $B8        in CHRGET-Zeiger
6152: 84 B9      261           STY  $B9        eintragen
                 262  *
                 263  * Programmende als LOMEM: für MOVE und LINKSET
6154: A5 AF      264           LDA  $AF
6156: 85 69      265           STA  $69
6158: A5 BØ      266           LDA  $BØ
615A: 85 6A      267           STA  $6A
615C: 2Ø 6C D6   268           JSR  $D66C      CLEAR aufrufen
                 269  *
615F: A5 1A      27Ø           LDA  ALTLEN     Alt$="" ?
6161: DØ Ø3      271           BNE  LINELOOP   Nein: los geht's
                 272  *
                 273  * Programm zu Ende oder Endzeile erreicht:
                 274  * nur noch Linkadressen neu berechnen, dann
                 275  * fertig (LINKSET macht automatisch Warmstart)
6163: 4C F2 D4   276  DONE     JMP  LINKSET
                 277  *
                 278  * Hauptschleife: $B8 zeigt auf Zeilenanfang
6166: AØ Ø1      279  LINELOOP LDY  #1
6168: B1 B8      28Ø           LDA  ($B8),Y    Programmende ?
616A: FØ F7      281           BEQ  DONE       Ja=< fertig
616C: C8         282           INY
616D: B1 B8      283           LDA  ($B8),Y    Zeilennummer
616F: 85 75      284           STA  $75        setzen
6171: C8         285           INY
6172: B1 B8      286           LDA  ($B8),Y
6174: 85 76      287           STA  $76
6176: A5 FC      288           LDA  ELINE      Endzeile kleiner
6178: C5 75      289           CMP  $75        als aktuelle ?
617A: A5 FD      29Ø           LDA  ELINE+1
617C: E5 76      291           SBC  $76
617E: 9Ø E3      292           BCC  DONE       Ja=< fertig
618Ø: A5 B8      293           LDA  $B8        Zeiger auf Zeile
```

```
6182: 85 DC     294          STA  ZPTR       merken
6184: A5 B9     295          LDA  $B9
6186: 85 DD     296          STA  ZPTR+1
6188: C8        297          INY  $B8        auf 1. Byte
6189: 84 D7     298          STY  SCHONDA    "Zeile nicht da"
618B: 20 98 D9  299          JSR  ADDY        setzen
618E: 20 A6 D9  300          JSR  $D9A6      Y:=Zeilenlänge-4
6191: 84 1E     301          STY  LEN        merken und als
6193: 84 D5     302          STY  RESTLEN    Abst. Zeilenende
                303  *       LDA   #0        nichts substrahieren
6195: 85 D0     304          STA  LAST       "neues Wort"
                305  *
                306  * ab nächstem Zeichen vergleichen:
6197: 49 FF     307  CHCKNEXT EOR #$FF       A von RestLen
6199: 38        308          SEC  abziehen
619A: 65 D5     309          ADC  RESTLEN    Zeilenrest
619C: 85 D5     310          STA  RESTLEN    kürzer als
619E: C5 1A     311          CMP  ALTLEN     Alt$ ?
61A0: B0 1A     312          BCS  COMPARE    Nein=< Vergleich
                313  *
                314  * Zeile kürzer als Alt$ oder zu lang:
61A2: 38        315  ZULANG2  SEC  $B8       auf nächste
61A3: 20 9A D9  316          JSR  ADDA        setzen
61A6: 4C 66 61  317          JMP  LINELOOP   und dort weiter
                318  *
                319  * Ab $B8 keine Übereinstimmung mit Alt$:
61A9: 20 4C 63  320  NOMATCH  JSR  CHKCTRC
61AC: A0 00     321          LDY  #0         hole dieses
61AE: B1 B8     322          LDA  ($B8),Y    Zeichen für
61B0: 85 D0     323          STA  LAST       Token-Search
61B2: A9 01     324          LDA  #1         Zeilenende rückt
61B4: E6 B8     325          INC  $B8        1 näher. $B8 um
61B6: D0 DF     326          BNE  CHCKNEXT   1 erhöhen und
61B8: E6 B9     327          INC  $B9        weiter
61BA: D0 DB     328          BNE  CHCKNEXT   (immer)
                329  *
                330  * Ab $B8 mit Alt$ vergleichen. Wenn Tokenmodus
                331  * und mitten im Wort, dann Alt$ nicht finden!
61BC: 20 4C 63  332  COMPARE  JSR  CHKCTRC
61BF: A0 00     333          LDY  #0
61C1: 24 D2     334          BIT  TOKEN      Token-Search ?
61C3: 10 0F     335          BPL  COMPLIT    Nein
61C5: A5 D0     336          LDA  LAST       letztes Zeichen
61C7: 20 2D 64  337          JSR  TOKENTEST  alphanumerisch ?
61CA: 90 08     338          BCC  COMPLIT    Nein=< OK
61CC: AD 01 02  339          LDA  ALT        fängt Alt$
61CF: 20 2D 64  340          JSR  TOKENTEST  alphanum. an ?
61D2: B0 D5     341          BCS  NOMATCH    Ja: nicht finden
61D4: 24 D3     342  COMPLIT  BIT  AUCHKLEI  V:=1, wenn Klein
61D6: B1 B8     343  COMPARE0 LDA  ($B8),Y   Zeichen holen
61D8: 70 0A     344          BVS  COMPGRKL   =< Klein><Groß
61DA: C9 61     345          CMP  #'a'       Groß/Klein egal:
61DC: 90 06     346          BCC  COMPGRKL   in Großbuch-
61DE: C9 7B     347          CMP  #'z'+1     staben wandeln
61E0: B0 02     348          BCS  COMPGRKL
61E2: 29 5F     349          AND  #$5F
61E4: D9 01 02  350  COMPGRKL CMP  ALT,Y     mit Alt$ vergl.
61E7: D0 C0     351          BNE  NOMATCH    =< Stimmt nicht!
61E9: C8        352          INY  Doch:      nächstes.
61EA: C4 1A     353          CPY  ALTLEN     Schon ganz Alt$?
61EC: 90 E8     354          BCC  COMPARE0   Nein=< weiter
61EE: 24 D2     355          BIT  TOKEN      Token-Search ?
61F0: 10 0C     356          BPL  COMPDLIT   Nein. Ja: endet
61F2: 20 2D 64  357          JSR  TOKENTEST  Alt$ alphanum.?
61F5: 90 07     358          BCC  COMPDLIT   Nein
61F7: B1 B8     359          LDA  ($B8),Y    folgendes Zchn.
61F9: 20 2D 64  360          JSR  TOKENTEST  alphanumerisch ?
61FC: B0 AB     361          BCS  NOMATCH    Ja: NICHT finden
                362  *
                363  * Alt$ wurde gefunden:
61FE: 20 66 62  364  COMPDLIT JSR  VERIFY    Ersetzen J/N ?
6201: B0 A6     365          BCS  NOMATCH    Nein!
6203: A5 1E     366          LDA  LEN        neue Zeilenlänge
6205: 65 1C     367          ADC  LENDIFF    berechnen
6207: 48        368          PHA
6208: 24 1D     369          BIT  DIFFVORZ   Zeile kürzer ?
620A: 30 06     370          BMI  DOMOVE     Ja=< immer OK
620C: B0 2D     371          BCS  ZULANG     zu lang=< Err
620E: C9 FA     372          CMP  #255-5     <249 Zeichen ?
6210: B0 29     373          BCS  ZULANG     Ja=< auch Err
                374  *
                375  * Zeilenlänge OK=< ab hier Programm
                376  * verschieben, wenn nötig, also die String-
                377  * längen unterschiedlich:
6212: 20 D6 62  378  DOMOVE   JSR  MOVE       Platz schaffen
                379  *
                380  * Programm nicht zu lang und Verschiebung OK:
```

```
6215: 68        381          PLA  nun        neue Zeilen-
6216: 85 1E     382          STA  LEN        länge eintragen
                383  * Wenn Neu$<>"",dann damit Alt$ überschreiben:
6218: A6 1B     384          LDX  NEULEN     Neu$="" ?
621A: F0 0D     385          BEQ  USEDNEW    Ja
621C: A0 00     386          LDY  #0         Nein: kopieren
621E: B1 08     387  USENEW  LDA  (NEU),Y    Byte holen
6220: 91 B8     388          STA  ($B8),Y    und eintragen
6222: C8        389          INY  Zeiger     erhöhen
6223: CA        390          DEX  alle       Zeichen ?
6224: D0 F8     391          BNE  USENEW     Nein=< weiterm.
6226: 20 98 D9  392          JSR  ADDY       $B8 dahinter
6229: 86 D0     393  USEDNEW STX  LAST       "neues Wort"
                394  *
                395  * Zeile auslisten, wenn nicht unterdrückt:
622B: 24 D4     396          BIT  NOLIST
622D: 30 07     397          BMI  DONTLIST
622F: 86 DA     398          STX  ZINV       weder INVERSE
6231: 86 DB     399          STX  ZNORM      noch NORMAL
6233: 20 3F 64  400          JSR  ZEILEHIN
                401  *
                402  * $B8 zeigt nun hinter den ersetzten Neu$.
                403  * Das Zeilenende rückt damit aber um die Länge
                404  * des Alt$ näher, da der ja ersetzt wurde!
6236: A5 1A     405  DONTLIST LDA ALTLEN
6238: 4C 97 61  406          JMP  CHCKNEXT   =< Schleife
                407  *
                408  * Zeile wurde zu lang:
623B: 68        409  ZULANG  PLA  Zeilenlänge weg
623C: 20 20 ED  410          JSR  $ED20      Zeilennummer
623F: A9 0B     411          LDA  #LNZULANG  und Text
6241: AC 00 60  412          LDY  STPAGE
6244: 20 3A DB  413          JSR  $DB3A      ausgeben
6247: 20 3A FF  414  ZULANG0 JSR  $FF3A      und Beep
624A: 20 42 63  415          JSR  KEYIN      Zeicheneingabe
624D: C9 45     416          CMP  #'E'       E=< Ende
624F: F0 0F     417          BEQ  QUIT
6251: C9 57     418          CMP  #'W'       W=< weitersuchen
6253: D0 F2     419          BNE  ZULANG0    nicht E/W=< Beep
6255: 20 5C DB  420          JSR  $DB5C      W: W ausgeben
6258: 20 8E FD  421          JSR  $FD8E      und CR
625B: A5 D5     422          LDA  RESTLEN    Restzeile über-
625D: 4C A2 61  423          JMP  ZULANG2    springen
6260: 20 5C DB  424  QUIT    JSR  $DB5C      E: E ausgeben
6263: 4C F2 D4  425          JMP  LINKSET    & Linkadressen
                426  *
                427  ** Unterprogramme für CHANGE **
                428  *
                429  * Ersetzen: J/N, wenn Some-Flag gesetzt:
6266: A5 D1     430  VERIFY  LDA  SOME
6268: 0A        431          ASL
6269: 90 38     432          BCC  VERIFIED
626B: A5 B8     433          LDA  $B8        Offset Alt$
626D: E5 DC     434          SBC  ZPTR        zum Zeilenanf.
626F: 85 DA     435          STA  ZINV       "Hier INVERSE"
6271: 18        436          CLC  +Alt$Länge
6272: 65 1A     437          ADC  ALTLEN
6274: 85 DB     438          STA  ZNORM      "Hier NORMAL"
6276: 20 3F 64  439          JSR  ZEILEHIN   Zeile listen
6279: A9 4F     440          LDA  #VFYTEXT   Frage ausgeben:
627B: AC 00 60  441          LDY  STPAGE
627E: 20 3A DB  442          JSR  $DB3A
6281: F0 03     443          BEQ  *+5
6283: 20 3A FF  444  VFYCMD  JSR  $FF3A      Beep
6286: 20 42 63  445          JSR  KEYIN
6289: C9 45     446          CMP  #'E'
628B: F0 D3     447          BEQ  QUIT       E=<Ende
628D: C9 41     448          CMP  #'A'       A=<ab hier alle
628F: D0 04     449          BNE  *+6
6291: 85 D1     450          STA  SOME       SOME löschen
6293: F0 08     451          BEQ  YES
6295: C9 4E     452          CMP  #'N'       N=<Nein
6297: F0 05     453          BEQ  NO
6299: C9 4A     454          CMP  #'J'
629B: D0 E6     455          BNE  VFYCMD
629D: 18        456  YES     CLC
629E: 08        457  NO      PHP
629F: 20 E1 64  458          JSR  ZEILEWEG
62A2: 28        459          PLP
62A3: 60        460  VERIFIED RTS
                461  *
                462  * Programm wurde zu lang:
62A4: A9 31     463  PZULANG LDA  #PGZULANG  Zeiger auf
62A6: AC 00 60  464          LDY  STPAGE     Fehlermeldung
62A9: 20 3A DB  465          JSR  $DB3A      String ausgeben
62AC: 20 19 ED  466          JSR  $ED19      IN Zeilennummer
62AF: 4C F2 D4  467          JMP  LINKSET    & Linkadressen
```

```
                468  *
                469  * Neu$ länger als Alt$=< nach hinten moven
62B2: A5 69     470  MOVEBACK LDA  $69        Programmende
62B4: 85 96     471           STA  $96        als altes Ende
62B6: 65 1C     472           ADC  LENDIFF    +Differenz
62B8: 85 94     473           STA  $94        als neues Ende
62BA: AA        474           TAX
62BB: A5 6A     475           LDA  $6A        Hi-Bytes
62BD: 85 97     476           STA  $97
62BF: 69 ØØ     477           ADC  #Ø
62C1: 85 95     478           STA  $95
62C3: C5 74     479           CMP  $74        Ende<HIMEM: ?
62C5: BØ DD     480           BCS  PZULANG    Ja: zu lang!
62C7: 86 69     481           STX  $69        Nein: Neues Ende
62C9: 85 6A     482           STA  $6A         eintragen
62CB: A5 B8     483           LDA  $B8        $B8 als alten
62CD: 85 9B     484           STA  $9B         Anfang
62CF: A5 B9     485           LDA  $B9
62D1: 85 9C     486           STA  $9C        jetzt moven mit
62D3: 4C 9A D3  487           JMP  $D39A      "BLTU2"
                488  *
                489  * ab $B8 LenDiff Bytes Platz schaffen/löschen:
62D6: A5 1C     490  MOVE     LDA  LENDIFF    gleich lang ?
62D8: FØ 32     491           BEQ  MOVED      Ja=< nichts tun
62DA: A5 1D     492           LDA  DIFFVORZ   Vorzeichen
62DC: ØA        493           ASL             ; in Carry
62DD: 9Ø D3     494           BCC  MOVEBACK   positiv
                495  *
                496  * Neu$ kürzer als Alt$ =< Bytes löschen:
62DF: A5 B8     497           LDA  $B8        $B8 als
62E1: 85 42     498           STA  $42         Zieladresse
62E3: E5 1C     499           SBC  LENDIFF    +Differenz als
62E5: 85 3C     5ØØ           STA  $3C         Quelladresse
62E7: A5 B9     5Ø1           LDA  $B9
62E9: 85 43     5Ø2           STA  $43
62EB: E9 FF     5Ø3           SBC  #-1
62ED: 85 3D     5Ø4           STA  $3D
62EF: A5 69     5Ø5           LDA  $69        Programmende+1
62F1: 85 3E     5Ø6           STA  $3E        als Endadresse
62F3: 18        5Ø7           CLC
62F4: 65 1C     5Ø8           ADC  LENDIFF    -Differenz
62F6: 85 69     5Ø9           STA  $69        =neues Programm-
62F8: A5 6A     51Ø           LDA  $6A         ende
62FA: 85 3F     511           STA  $3F
62FC: 69 FF     512           ADC  #-1
62FE: 85 6A     513           STA  $6A
63ØØ: AØ ØØ     514           LDY  #Ø
63Ø2: 4C 2C FE  515           JMP  $FE2C      Monitor-MOVE
                516  *
                517  * Zeilennummer in ASCII ab $B8 nach X/Y:
63Ø5: 2Ø ØC DA  518  INNUM    JSR  $DAØC      ROM-Routine
63Ø8: A6 5Ø     519           LDX  $5Ø        X/Y mit Ergebnis
63ØA: A4 51     52Ø           LDY  $51        der Routine
63ØC: 6Ø        521  MOVED    RTS  laden
                522  *
                523  * Mode-Zeichen auswerten:
                524  * T        - Token ein, Literal aus
                525  * L        - Literal ein, Token aus
                526  * K        - Groß-/Kleinbuchstaben unterschdn.
                527  * S oder V - Ersetzen: J/N einschalten
                528  * U        - Zeilenauflistung unterdrücken
                529  * Ctrl-C   - BREAK und Ende
63ØD: 2Ø B1 ØØ  53Ø  MODEINØ  JSR  CHRGET     nächstes Zeichen
631Ø: C9 Ø3     531  MODEINPUT CMP  #3         ↑C=< BREAK
6312: FØ 55     532           BEQ  CTRC
6314: 29 5F     533           AND  #$5F       Großbuchstabe
6316: C9 55     534           CMP  #'U'       Unterdrücken ?
6318: DØ Ø4     535           BNE  KEINUNTDR
631A: 66 D4     536           ROR  NOLIST
631C: 3Ø EF     537           BMI  MODEINØ
631E: C9 4B     538  KEINUNTDR CMP  #'K'       K-Option ?
632Ø: DØ Ø4     539           BNE  KEINKLEI   Nein
6322: 66 D3     54Ø           ROR  AUCHKLEI   Ja: setzen
6324: 3Ø E7     541           BMI  MODEINØ    und weiter
6326: C9 53     542  KEINKLEI CMP  #'S'       Some (EDASM) ?
6328: FØ Ø4     543           BEQ  ONLYSOME    oder
632A: C9 56     544           CMP  #'V'       oder Verify ?
632C: DØ Ø4     545           BNE  NOTSOME     (UCSD)
632E: 66 D1     546  ONLYSOME ROR  SOME       Ja: setzen
633Ø: 3Ø DB     547           BMI  MODEINØ
6332: C9 54     548  NOTSOME  CMP  #'T'       Token-Search ?
6334: DØ Ø4     549           BNE  NOTTOKEN
6336: 66 D2     55Ø           ROR  TOKEN      Ja: setzen
6338: 3Ø D3     551           BMI  MODEINØ
633A: 49 4C     552  NOTTOKEN EOR  #'L'       auch nicht L ?
633C: DØ CE     553           BNE  MOVED      Nein=< Delim folgt
633E: 85 D2     554           STA  TOKEN      Doch: kein Token
6341: FØ CB     555           BEQ  MODEINØ    und weiter
                556  *
                557  * Zeicheneingabe. Wenn Ctrl-C, dann BREAK IN:
6342: 2Ø ØC FD  558  KEYIN    JSR  $FDØC
6345: C9 83     559           CMP  #$83
6347: FØ ØD     56Ø           BEQ  LINKCTRC
6349: 29 5F     561           AND  #$5F
634B: 6Ø        562  NOCTRC   RTS
                563  *
                564  * Tastatur auslesen. Wenn Ctrl-C, BREAK IN:
634C: 2C 3Ø CØ  565  CHKCTRC  BIT  $CØ3Ø      Ausstiegsknack
634F: AD ØØ CØ  566           LDA  $CØØØ      KBD
6352: C9 83     567           CMP  #$83       Ctrl-C ?
6354: DØ F5     568           BNE  NOCTRC     Nein=< RTS
                569  *
                57Ø  * Ctrl-C gedrückt. BREAK IN ausgeben und
                571  * Linkadressen neu berechnen (danach!)
6356: 2Ø 5C 63  572  LINKCTRC JSR  FAKECTRC   BREAK IN Zeile
6359: 4C F2 D4  573           JMP  LINKSET
                574  *
                575  * BREAK IN Zeilennummer ausgeben (simulieren):
635C: 8D 1Ø CØ  576  FAKECTRC STA  $CØ1Ø      Taste löschen
635F: A9 5D     577           LDA  #$D35D     Zeiger auf
6361: AØ D3     578           LDY  #>$D35D     BREAK+Beep
6363: 2Ø 3A DB  579           JSR  $DB3A      Text und
6366: 4C 19 ED  58Ø           JMP  $ED19      Zeilennr. ausg.
                581  *
                582  * Ctrl-C als Delimiter =< BREAK:
6369: 4C 63 D8  583  CTRC     JMP  $D863
                584  *
                585  * Unzulässiger Delimiter! Error & Warmstart:
636C: A9 74     586  DELIMERR LDA  #INVDELIM  Zeiger auf Text
636E: AC ØØ 6Ø  587           LDY  STPAGE
6371: 2Ø 3A DB  588           JSR  $DB3A      Text ausgeben
6374: 4C 2A D4  589           JMP  $D42A      weiter wie Error
                59Ø  *
                591  * Alt$ und Neu$ bestimmen. $B8 zeigt auf
                592  * Zeichen, das als Delimiter benutzt wird:
6377: 2Ø B7 ØØ  593  GETSTRGS JSR  CHRGOT     Delimiter holen
637A: 85 FE     594           STA  DELIM       und merken
637C: 2Ø 2D 64  595           JSR  TOKENTEST  alphanum. ?!?
637F: BØ EB     596           BCS  DELIMERR   Ja=< Error!
6381: C9 Ø3     597           CMP  #3         Ctrl-C ?
6383: FØ E4     598           BEQ  CTRC       =< BREAK
6385: 9Ø E5     599           BCC  DELIMERR   darunter=< Error
                6ØØ  * Delimiter steht fest. X als Zeiger auf End-
                6Ø1  * delimiter des Alt$ bestimmen. Dabei Klein-
                6Ø2  * in Großbuchstaben wandeln. Wenn KEINE
                6Ø3  * Groß/Klein-Unterscheidung gewünscht, dann
                6Ø4  * auch zwischen Anführungszeichen!
6387: A6 B8     6Ø5           LDX  $B8        Zeiger holen
6389: 24 D3     6Ø6           BIT  AUCHKLEI   V:=1, wenn Klein
638B: 2Ø F6 63  6Ø7           JSR  FINDEND    Enddelim. holen
                6Ø8  * Wird Alt$ durch Zeilenende beendet, dann
                6Ø9  * SYNTAX ERROR. Sonst Zeiger merken:
638E: FØ 42     61Ø           BEQ  SYNERR
639Ø: 86 1E     611           STX  LEN
                612  * gegebenen Neu$ um 3 Bytes nach hinten ver-
                613  * schieben, damit er beim Kodieren des Alt$
                614  * nicht zerstört wird! Alt$-End-Delimiter als
                615  * Neu$-Start-Delimiter mitverschieben:
6392: AØ FC     616           LDY  #252
6394: B9 ØØ Ø2  617  GETSTRGSØ LDA  $2ØØ,Y
6397: 99 Ø3 Ø2  618           STA  $2ØØ+3,Y
639A: 88        619           DEY
639B: C4 1E     62Ø           CPY  LEN
639D: BØ F5     621           BCS  GETSTRGSØ
                622  * Nun Alt$-End-Delimiter durch Ø ersetzen, um
                623  * Kodier-Routine Zeilenende vorzuspielen:
639F: A9 ØØ     624           LDA  #Ø
63A1: 9D ØØ Ø2  625           STA  $2ØØ,X
63A4:           626  * Zeiger auf Anfang des Neu$ in LEN merken:
63A4: E8        627           INX
63A5: E8        628           INX
63A6: E8        629           INX
63A7: 86 1E     63Ø           STX  LEN        Delimiter
63A9: E8        631           INX
63AA: 86 Ø8     632           STX  NEU        1. Zeichen
63AC: A9 Ø2     633           LDA  #>$2ØØ     Neu$ steht auch
63AE: 85 Ø9     634           STA  NEU+1       in Page 2
                635  * Nun Alt-String mit ROM-Routine kodieren:
63BØ: 2Ø 59 D5  636           JSR  $D559      mit "GETIN"
                637  * jetzt zeigt $2ØØ-5,Y auf die neue End-Ø
63B3: 98        638           TYA
63B4: 38        639           SEC
63B5: E9 Ø6     64Ø           SBC  #6
                641  * und nun ist A+$2Ø1 die Adresse des End-
```

Ausgabe und Eingabe mit TYPETERM®

im Slot Ihres

APPLE II/IIe

Das bedeutet: Computertextverarbeitung von der Schreibmaschinentastatur! Steckerfertig ohne Umbau.

Die neue CE-550! mit TYPETERM DM 1.398,–

TYPETERM- DM 479,–
Interface
für alle BROTHER-Typenradschreibmaschinen ab AX-30 bis EM-811
(auch für Vorgängermodelle!)
Paketpreis z. B.:

EM-501 mit TYPETERM ..... DM 2136,–
EM-511 mit TYPETERM ..... DM 2412,–
EM-701 mit TYPETERM ..... DM 2468,–

TYPETERM – ein starkes Interface für starke Maschinen! Alle Cursor- und Ctl-Befehle, 4k ROM auf der Karte für DOS, PRODOS, CP/M, PASCAL, 2 Zeichensätze verfügbar z. B. deutsch u. ASCII. Alle Features: Hoch-/Tiefstellen, autom. Unterstreichen, var. Zeichen und Zeilenabst., autom. Papierzuführung usw.

TYPETERM – ein Produkt von

interkom electronic Kock & Mreches GmbH Postf., 3004 Isernhagen 4 Telefon 05139-87393

Ausgabe mit TYPETERM® JUNIOR

im Slot Ihres

APPLE II/IIe

Paketpreis **DM 899,–**
Schreibmaschine AX-10 mit Interface TYPETERM JUNIOR, steckfertig.

brother
Die Zukunft heute

TYPETERM JUNIOR mit AX-10 – unser besonders günstiges Gespann, ebenfalls steckfertig. Mit TYPETERM JUNIOR kann die AX-10 mehr. Sie wird zum vollwertigen Typenraddrucker für Ihren Apple:

- 3 verschiedene Schriftstärken
- Automatisches Unterstreichen
- 2 Zeichensätze z.B. deutsch u. ASCII
- 2 Zeichenabstände
- 2k ROM auf der Karte für Ausgabe unter DOS, PRODOS, CP/M u. PASCAL.

TYPETERM JUNIOR – ein Produkt von

interkom electronic Kock & Mreches GmbH Postf., 3004 Isernhagen 4 Telefon 05139-87393

Es gehört inzwischen zum Standard für ein modernes Rechnersystem, mit einer Festplatte ausgerüstet zu sein. Erst dadurch erlangt der Rechner die Qualität in der Datenverarbeitung, wie sie bei professionellen Anwendungen verlangt wird.

Beim Apple wird einfach das Netzteil herausgenommen und dafür das 10/20 MByte MEGA-CORE eingesetzt. Ab dann warten vier Betriebssysteme (DOS, CP/M, UCSD-Pascal, ProDOS) auf Ihr Kommando. Welcher Profirechner kann das schon?

Fragen Sie uns nach Fakten, Preisen, Bezugsquellen und holen Sie sich für 5,-- DM unsere Demo-Diskette.

Ein Produkt von:

FRANK & BRITTING
Elektronik Entwicklungs GmbH
Lange Straße 4, 7529 Forst
Telefon: 07251 /103068-69
Telex: 7822452 fub d

Die Harddiskcontroller-Spezialisten

Apple und IBM kompatible Computer

16K, Z80, Diskcontroller je 98,–
80 Zeichenkarte mit Softswitch
2 Zeichensätze 198,–
IIe-kompatibles Motherboard
ohne Firmware 498,–
Erphi-controller mit Autopatch . . 285,–
TEAC FD-54A mit Apple-Bus . . . 298,–
PC II+ Karte läßt alle Apple II+ Software auf dem IBM zu. Deutsche Entwicklung und Fertigung . 1175,–
OLYMPIA compact NP 1298,–
Tastaturen für IBM und Apple . ab 298,–
512K-RAM-Karte mit 256 K bestückt inkl. Software . . 298,–
Apple Super-Modem-Karte inkl. dt. Software und dt. Handbuch . . . 348,–
Versand nur per Nachnahme oder Vorkasse
Weiteres Zubehör für Apple und IBM gegen frankierten Rückumschlag.
128K Karte (Saturn kompatibel) . **278,–**

MoVe GmbH
vormals U. Mohwinkel Electronic
Berliner Straße 73 Pf: 250 166
5090 Leverkusen Fettehenne
Telefon 02 14/9 37 81 od. 9 50 60

M2000
IIc-ProDOS-Uhr

- **Anschluß an den seriellen Port 2 des Apple-IIc**
- **kompatibel mit Appleworks, BASIC. SYSTEM, EDASM.SYSTEM, MERLIN.SYSTEM u. a.**
- **programmierbar als selbständige Schaltuhr**
- **acht digitale 100 mA-Ausgänge**
- **acht digitale CMOS-Eingänge**
- **inclusiv Programmdiskette für Apple-IIc (Demo, Utilities)**

M2000: 387,– DM
V.24-Kabel für IIc: 67,– DM

IDW – ELECTRONIC
Max-Anderl-Straße 109, 8056 Neufahrn
Telefon 0 81 65 - 36 41

Platinenvorlagen einfach mit dem Computer erstellen!!!

LAYOUT-B
Computerpaket, die Komplettlösung für Profis mit CAD 1280 EL Software zur komfortablen Platinen und Schaltungserstellung mit Linerouter und Gummibandmethode. Komplettsystem mit 512 k RAM, 2-Disk, hochauflösende Grafikkarte, Farbmonitor, Plotter und Grafiktablett.
Setpreis nur ..... **DM 14100,–**
CAD 1280 EL Software allein, für IBM ..... DM 1940,–

LAYOUT-A
Computerpaket, die Komplettlösung für Praktiker, bestehend aus: Computer 64k mit 2 Disklaufwerken und abgesetzter Tastatur, Monitor 12 Zoll, grün oder bernstein, Plotter DIN A3 Format,4 Farben, Software, Geräte getestet.
Setpreis nur ..... **DM 5885,–**
LAYOUT-A, Software für APPLE-II-plus ..... DM 985,–

Hochauflösende Grafikkarten
1024 × 1024 bis 2048 × 2048 Punkte BildRAM, bis 16 Farben mit 7220 Grafikcontroller. Bildschirmformat und Frequenz programmierbar. Zeichnet blitzschnell Punkte, Linien, Kreis, Rechteck, Schrift, Zoom, PAN usw. Monitorauflösung z.B. 1024 × 567 Pixel. Slotsteckkarten für IBM, ECB-Bus, APPLE II ..... ab 780.– DM

Plotter
A3 incl. Treibersoft, 0,05mm ..... nur 2198,– DM

Info anfordern bei:
Klaus G. Gorny
Computer Soft & Hardware
3131 Schweskau Nr. 28, Tel. 05883/1293

Bühler 16-BIT Hit

IBM komp. 16 BIT Computer incl. Turbo 640 K – 6/8 MHz · Multi I/O · Herkules · 150 W Netzteil · Deutsch. Keyboard · Floppy 2×40 Track **BN 64036 1398,–**

IBM komp. Keyboard. Deutsche Tastatur. **BN 91245 198,–**
ab 5 Stck. **BN 91246 178,–**

IBM komp. Floppy Slim Line. 2×40 Track/360 KB. 1 Jahr Austausch-Garantie. **BN 64019 258,–**

IBM komp. Floppy Full High 2×40 Track/360 KB Restposten. Geprüft, ½ Jahr Garantie. Voller Service garantiert.
BN 97175 158,—
ab 4 Stck. **BN 97176 149,80**

Kostenlos Katalog anfordern!

Bühler Copmputer: Postfach 32, 7570 Baden-Baden ★ Shop: Waldstraße 46, 7500 Karlsruhe

```
                642  * Delimiters =< A=Länge von Alt$
63B7: 85 1A     643          STA  ALTLEN     eintragen
                644  * Nun das Ende des Neu$ suchen. Innerhalb von
                645  * Anführungszeichen grundsätzlich keine Groß-
                646  * buchstaben-Umwandlung!
63B9: A6 1E     647          LDX  LEN
63BB: 2C A8 65  648          BIT  SEV        V:=1!
63BE: 20 F6 63  649          JSR  FINDEND    suchen
                650  * Wenn Neu$ nicht mit Zeilenende endet, dann
                651  * prüfen, ob danach noch Zeichen folgen. Wenn
                652  * ja, dann SYNTAX ERROR, sonst End-Delimiter
                653  * durch Ø als Zeilenendmarke ersetzen:
63C1: FØ 12     654          BEQ  CODIERE2   Ja
63C3: A9 ØØ     655          LDA  #Ø         Ø zeigt Zeilen-
63C5: 9D ØØ Ø2  656          STA  $2ØØ,X      ende an
63C8: 86 B8     657          STX  $B8        $B8 setzen
63CA: E6 B9     658          INC  $B9        =< $B9 enthält 2
63CC: 2Ø B1 ØØ  659          JSR  CHRGET     Zeichen danach
63CF: A8        660          TAY  Zeilenende ?
63DØ: FØ Ø3     661          BEQ  CODIERE2   Ja
63D2: 4C C9 DE  662  SYNERR  JMP  $DEC9      Nein=< Err!
                663  * Neu$ ab $2ØØ+LEN nach $2ØØ+LEN kodieren:
63D5: 85 13     664  CODIERE2 STA $13        lösche DATA-Flag
63D7: 85 1D     665          STA  DIFFVORZ   schon mal auf Ø
63D9: A5 1E     666          LDA  LEN
63DB: AA        667          TAX             ab hier kodieren
63DC: 18        668          CLC
63DD: 69 Ø4     669          ADC  #4
63DF: A8        670          TAY             ab hier ablegen
63EØ: 84 1E     671          STY  LEN        Start merken
63E2: 2Ø 6D D5  672          JSR  $D56D      kodieren
                673  * Länge des Neu$ berechnen:
63E5: 88        674          DEY  Zeiger     auf End-Ø
63E6: 98        675          TYA  -1
63E7: 18        676          CLC  -1
63E8: E5 1E     677          SBC  LEN        -(Startzeiger-1)
63EA: 85 1B     678          STA  NEULEN     =Länge von Neu$
                679  * Differenz bestimmen (DIFFVORZ schon Ø):
63EC: 38        680          SEC
63ED: E5 1A     681          SBC  ALTLEN     Neue-Alte Länge
63EF: 85 1C     682          STA  LENDIFF     merken
63F1: BØ Ø2     683          BCS  *+4        Neu$ länger ?
63F3: C6 1D     684          DEC  DIFFVORZ   Nein=< negativ
63F5: 6Ø        685          RTS
                686  *
                687  * ab $2Ø1,X nach Stringende suchen und Klein-
                688  * buchstaben in große wandeln. Wenn V=Ø, dann
                689  * auch innerhalb von Anführungszeichen:
63F6: A4 FE     690  FINDEND LDY  DELIM
63F8: A9 ØØ     691          LDA  #Ø
63FA: FØ Ø4     692          BEQ  FINDDELIM
63FC: A4 ØD     693  FINDQUOTE LDY $ØD       Delim und Ø
63FE: A5 ØE     694          LDA  $ØE        vertauschen
64ØØ: 84 ØE     695  FINDDELIM STY $ØE       wird verglichen
64Ø2: 85 ØD     696          STA  $ØD        der nicht
64Ø4: E8        697  FINDENDØ INX nächstes   Zeichen
64Ø5: BD ØØ Ø2  698          LDA  $2ØØ,X      holen
64Ø8: FØ 21     699          BEQ  FOUNDEND   Zeilenende
64ØA: 5Ø Ø8     7ØØ          BVC  FINDKLEI   Klein ?
64ØC: CØ ØØ     7Ø1          CPY  #Ø         Ja: in "" ?
64ØE: FØ 11     7Ø2          BEQ  FINDEND1   Ja
641Ø: CØ 22     7Ø3          CPY  #'"'       gesamt in "" ?
6412: FØ ØD     7Ø4          BEQ  FINDEND1   Ja
6414: C9 61     7Ø5  FINDKLEI CMP #'a'       Nein: Klein ?
6416: 9Ø Ø9     7Ø6          BCC  FINDEND1
6418: C9 7B     7Ø7          CMP  #'z'+1
641A: BØ Ø5     7Ø8          BCS  FINDEND1   Nein
641C: 29 5F     7Ø9          AND  #$5F       Ja: wandeln
641E: 9D ØØ Ø2  71Ø          STA  $2ØØ,X     & eintragen
6421: C5 ØE     711  FINDEND1 CMP $ØE        Delimiter ?
6423: FØ Ø6     712          BEQ  FOUNDEND   Ja=< fertig
6425: C9 22     713          CMP  #'"'       Anfang/Ende von
6427: FØ D3     714          BEQ  FINDQUOTE  ""=< vertauschen
6429: DØ D9     715          BNE  FINDENDØ   sonst weiter
642B: A8        716  FOUNDEND TAY Ø          (EQ) oder
642C: 6Ø        717          RTS  Delimiter  (NE) ?
                718  *
                719  * Testroutine für Token-Search. C=Ø, wenn
                72Ø  * weder Ziffer noch Buchstabe:
642D: C9 3Ø     721  TOKENTEST CMP #'Ø'
642F: 9Ø ØA     722          BCC  SPECIAL
6431: C9 3A     723          CMP  #'9'+1
6433: 9Ø Ø5     724          BCC  ALPHANUM
6435: 29 DF     725          AND  #$DF
6437: 4C 7D EØ  726          JMP  $EØ7D
643A: 38        727  ALPHANUM SEC
```

```
643B: 6Ø        728  SPECIAL  RTS
                729  *
                73Ø  * Zeile ab ZPTR listen. Byte ZINV bis ZNORM-1
                731  * in INVERSE ausgeben:
643C: 2Ø E1 64  732  ZEILEHINØ JSR  ZEILEWEG   Schon da: weg
643F: 24 D7     733  ZEILEHIN BIT  SCHONDA
6441: 3Ø F9     734          BMI  ZEILEHINØ
6443: 2Ø 57 DB  735          JSR  $DB57      Space
6446: 2Ø 2Ø ED  736          JSR  $ED2Ø      Zeilennummer
6449: 2Ø 57 DB  737          JSR  $DB57      Space
644C: A2 ØØ     738          LDX  #Ø         nullsetzen:
644E: 86 DF     739          STX  LINCOUNT   Zeilenzähler &
645Ø: 86 9D     74Ø          STX  $9D        Tokenzeiger Lo
6452: CA        741          DEX
6453: 86 D7     742          STX  SCHONDA    "Zeile da"
6455: AØ Ø4     743          LDY  #4
6457: 84 85     744  HINØ    STY  $85        Y merken
6459: C4 DA     745          CPY  ZINV
645B: DØ Ø6     746          BNE  *+8
645D: 2Ø 8Ø FE  747          JSR  $FE8Ø      Inverse
646Ø: 4C 6A 64  748          JMP  HIN1
6463: C4 DB     749          CPY  ZNORM
6465: DØ Ø3     75Ø          BNE  HIN1
6467: 2Ø 84 FE  751          JSR  $FE84      NORMAL
646A: 2Ø CØ 64  752  HIN1    JSR  CHKCR      CR, wenn nötig
646D: A4 85     753          LDY  $85        Zeichen holen
646F: B1 DC     754          LDA  (ZPTR),Y
6471: FØ 37     755          BEQ  LISTED     Ø=< Zeilenende
6473: 1Ø 2Ø     756          BPL  HIN2       ASCII: ausgeben
6475: AA        757          TAX  Token:
6476: 2Ø 57 DB  758          JSR  $DB57      Space
6479: AØ CF     759          LDY  #$DØDØ-1   Token suchen
647B: A9 DØ     76Ø          LDA  #>$DØDØ-1  (Tokenzeiger Hi
647D: 85 9E     761          STA  $9E         setzen)
647F: CA        762  GETTOK  DEX
648Ø: 1Ø Ø7     763          BPL  OUTTOK
6482: 2Ø 2C D7  764  SKIPTOK JSR  $D72C      nächstes Zeichen
6485: 1Ø FB     765          BPL  SKIPTOK    bis Token-Ende
6487: 3Ø F6     766          BMI  GETTOK     weiterzählen
6489: 2Ø 2C D7  767  OUTTOK  JSR  $D72C      nächstes Zeichen
648C: AA        768          TAX
648D: 2Ø 5C DB  769          JSR  $DB5C      ausgeben
649Ø: 8A        77Ø          TXA  bis
6491: 1Ø F6     771          BPL  OUTTOK     Token-Ende
6493: A9 2Ø     772          LDA  #32        Space
6495: C9 2Ø     773  HIN2    CMP  #32        Control-Zeichen
6497: BØ Ø9     774          BCS  HIN3       Nein
6499: AA        775          TAX  Ja:        könnte Bild-
649A: A9 5E     776          LDA  #'↑'       aufbau durchein-
649C: 2Ø 5C DB  777          JSR  $DB5C      ander bringen
649F: 8A        778          TXA  =<         mit ↑ als
64AØ: Ø9 4Ø     779          ORA  #$4Ø       Buchstabe
64A2: 2Ø 5C DB  78Ø  HIN3    JSR  $DB5C      ausgeben
64A5: A4 85     781          LDY  $85        (auch Tokens)
64A7: C8        782          INY  nächsten
64A8: DØ AD     783          BNE  HINØ       und weiter
                784  *
                785  * fertig gelistet: NORMAL & CR:
64AA: 2Ø 84 FE  786  LISTED  JSR  $FE84      NORMAL
                787  *
                788  * CR ausgeben und mitzählen:
64AD: A5 32     789  OUTCR   LDA  $32        INVFLG
64AF: 48        79Ø          PHA  retten
64BØ: 2Ø 84 FE  791          JSR  $FE84      NORMAL (Scroll?)
64B3: 2Ø 8E FD  792          JSR  $FD8E      CR
64B6: 68        793          PLA  altes      INVFLG
64B7: 85 32     794          STA  $32
64B9: E6 DF     795          INC  LINCOUNT   Zeilenzähler
64BB: 1Ø Ø2     796          BPL  *+4        (muß kleiner
64BD: C6 DF     797          DEC  LINCOUNT   als 128 bleiben)
64BF: 6Ø        798          RTS
                799  *
                8ØØ  * CR ausgeben und mitzählen:
64CØ: AD Ø6 EØ  8Ø1  CHKCR   LDA  $EØØ6      BASIC-Version
64C3: FØ ØE     8Ø2          BEQ  CHKCR4Ø    II+, Alt-//e
64C5: 2C 1F CØ  8Ø3          BIT  $CØ1F      8Ø Zeichen
64C8: 1Ø Ø9     8Ø4          BPL  CHKCR4Ø    Nein
64CA: AD 7B Ø5  8Ø5          LDA  $57B       Neu-//e, //c
64CD: C9 49     8Ø6          CMP  #8Ø-7      am Rand ?
64CF: 9Ø ØF     8Ø7          BCC  CHKEDCR    Nein
64D1: BØ Ø6     8Ø8          BCS  CHKCR8Ø    Ja
64D3: A5 24     8Ø9  CHKCR4Ø LDA  $24        4Ø Zeichen:
64D5: C9 21     81Ø          CMP  #4Ø-7      am Rand ?
64D7: 9Ø Ø7     811          BCC  CHKEDCR    Nein
64D9: 2Ø AD 64  812  CHKCR8Ø JSR  OUTCR      Ja: CR
64DC: A9 Ø5     813          LDA  #6-1       und
64DE: 85 24     814          STA  $24        HTAB 6
```

```
64E0: 60          815  CHKEDCR  RTS
                  816  *
                  817  * Zeile wieder löschen:
64E1: 24 D7       818  ZEILEWEG BIT  SCHONDA     Zeile überhaupt
64E3: 10 FB       819           BPL  CHKEDCR     da ? Nein:fertig
64E5: 46 D7       820           LSR  SCHONDA     "nicht da"
64E7: A0 00       821  WEG      LDY  #0          Zeile ab Spalte
64E9: 20 9E FC    822           JSR  $FC9E       0 löschen
64EC: 20 1A FC    823           JSR  $FC1A       Cursor 1 hoch
64EF: C6 DF       824           DEC  LINCOUNT    Zeilenzähler
64F1: 10 F4       825           BPL  WEG         noch nicht alle
64F3: 4C 8E FD    826           JMP  $FD8E       alle: CR
                  827  *
                  828  *** REMS: REM-Spacekiller **
                  829  *
64F6: 20 17 9A    830  KILLSPACE JSR  $9A17       BASIC.SYSTEM an
64F9: A5 67       831           LDA  $67
64FB: A4 68       832           LDY  $68
64FD: 85 B8       833           STA  WOHER       ab Programmstart
64FF: 84 B9       834           STY  WOHER+1     durchsuchen
6501: 85 CE       835           STA  WOHIN       und dort ablegen
6503: 84 CF       836           STY  WOHIN+1
6505: 46 D0       837           LSR  BRKFLAG     normal laufen
6507: 10 03       838           BPL  LOOP
                  839  *
                  840  * Zeilenende erreicht=< bis dahin übernehmen:
6509: 20 8F 65    841  KILLEND  JSR  COPY        übernehmen
650C: A0 01       842  LOOP     LDY  #1
650E: B1 B8       843           LDA  (WOHER),Y   Programmende ?
6510: F0 48       844           BEQ  KILLDONE    Ja=< Schluß
6512: AD 00 C0    845           LDA  $C000       Tastatur lesen
6515: C9 83       846           CMP  #$83        Ctrl-C ?
6517: D0 13       847           BNE  KEINCTRC    Nein
6519: 24 D0       848           BIT  BRKFLAG     war schon BREAK?
651B: 30 0F       849           BMI  KEINCTRC    Ja
651D: 85 D0       850           STA  BRKFLAG     Nein: BRK-Flag
651F: C8          851           INY               setzen, d.h.
6520: B1 B8       852           LDA  (WOHER),Y    von nun an
6522: 85 75       853           STA  $75          nichts mehr killen
6524: C8          854           INY
6525: B1 B8       855           LDA  (WOHER),Y
6527: 85 76       856           STA  $76
6529: 20 5C 63    857           JSR  FAKECTRC    BREAK IN
652C: A0 04       858  KEINCTRC LDY  #4          start mit 4.Byte
652E: 2C 30 C0    859  KILLLOOP BIT  $C030       Knack
6531: B1 B8       860           LDA  (WOHER),Y   Byte holen
6533: F0 D4       861           BEQ  KILLEND     0=< Zeilenende
6535: C9 B2       862           CMP  #$B2        REM ?
6537: F0 07       863           BEQ  KILLREM     Ja
6539: C9 83       864           CMP  #$83        oder DATA ?
653B: F0 0E       865           BEQ  KILLDATA    Klar!
653D: C8          866  KILLNEXT INY              nächstes Byte
653E: D0 EE       867           BNE  KILLLOOP    und in Schleife
                  868  *
                  869  * REM gefunden: erstmal bis REM einschließlich
                  870  * übernehmen, Leerzeichen mit SKIPSPC über-
                  871  * springen, Zeilenende suchen und bis dahin
                  872  * übernehmen:
6540: 20 8F 65    873  KILLREM  JSR  COPY        übernehmen
6543: 20 88 65    874           JSR  SKIPSPC
6546: 20 A6 D9    875           JSR  $D9A6       Zeilenende
6549: F0 BE       876           BEQ  KILLEND     (immer)
                  877  *
                  878  * DATA gefunden: bis dahin übernehmen. Dann
                  879  * führende Leerzeichen ignorieren
                  880  * und Ende des nächsten Daten-Feldes suchen.
                  881  * Endet es mit Komma, dann weiter, als wenn
                  882  * DATA gefunden. Sonst zurück in Schleife:
654B: 20 8F 65    883  KILLDATA JSR  COPY        übernehmen
654E: 20 88 65    884           JSR  SKIPSPC
6551: 20 6B 65    885           JSR  FIELD       Datenfeld-Ende
6554: C9 2C       886           CMP  #','        Komma=< weitere
6556: F0 F3       887           BEQ  KILLDATA     Felder folgen
6558: D0 D4       888           BNE  KILLLOOP    sonst Schleife
                  889  *
                  890  * fertig:
655A: 20 8F 65    891  KILLDONE JSR  COPY        Abschluß-Nullen
655D: A5 CE       892           LDA  WOHIN       WOHIN=Ende
655F: A4 CF       893           LDY  WOHIN+1     also übernehmen
6561: 85 69       894           STA  $69         als LOMEM:
6563: 84 6A       895           STY  $6A         für LINKSET
6565: 4C F2 D4    896           JMP  LINKSET     Linkadressen
                  897  *
                  898  * 1. Byte im Datenfeld suchen, d.h. Null,
                  899  * Doppelpunkt oder Komma, letztere nur, wenn
                  900  * nicht innerhalb von Anführungszeichen:
6568: D0 1A       901  IMSTRG   BNE  FIELD1      kein '"'
656A: C8          902           INY              doch
```

```
656B: B8          903  FIELD    CLV              ("nicht Strg")
656C: B1 B8       904  FIELD0   LDA  (WOHER),Y   Zeichen holen
656E: F0 17       905           BEQ  FIELDDONE   0=< Ende
6570: C9 22       906           CMP  #'"'        ist '"' ?
6572: 70 F4       907           BVS  IMSTRG      (im String)
6574: D0 06       908           BNE  KEINSTBEG   Stringanfang ?
6576: 2C A8 65    909           BIT  SEV         Ja: "im Strg"
6579: C8          910           INY
657A: 70 F0       911           BVS  FIELD0      und weiter
657C: C9 2C       912  KEINSTBEG CMP  #','        außerhalb: ',' ?
657E: F0 07       913           BEQ  FIELDDONE   Ja=< fertig
6580: C9 3A       914           CMP  #':'        oder ':' ?
6582: F0 03       915           BEQ  FIELDDONE   Ja=< auch fertig
6584: C8          916  FIELD1   INY               Nein=< nächstes
6585: D0 E5       917           BNE  FIELD0      Zeichen
6587: 60          918  FIELDDONE RTS
                  919  *
                  920  * Leerzeichen überspringen, wenn nicht schon
                  921  * Ctrl-C gedrückt wurde:
6588: 24 D0       922  SKIPSPC  BIT  BRKFLAG
658A: 30 FB       923           BMI  FIELDDONE
658C: 4C B7 00    924           JMP  CHRGOT
                  925  *
                  926  * von (WOHER) bis (WOHER)+Y übernehmen,
                  927  * WOHER dahinter setzen und WOHIN um dieselbe
                  928  * Anzahl erhöhen:
658F: 98          929  COPY     TYA  X:=Y+1      als Byte-
6590: AA          930           TAX  Zähler      (1..256
6591: E8          931           INX  Bytes)
6592: A0 00       932           LDY  #0
6594: B1 B8       933  COPY0    LDA  (WOHER),Y   (mind. 1 Byte)
6596: 91 CE       934           STA  (WOHIN),Y   kopieren
6598: C8          935           INY
6599: CA          936           DEX
659A: D0 F8       937           BNE  COPY0
659C: A2 B8       938           LDX  #WOHER      WOHER um ehem.
659E: 20 A9 65    939           JSR  ADD          Y+1 erhöhen,
65A1: A2 CE       940           LDX  #WOHIN      WOHIN auch
65A3: 20 A9 65    941           JSR  ADD
65A6: A0 00       942           LDY  #0          und Y nullsetzen
65A8: 60          943  SEV      RTS
                  944  *
                  945  * 16-Bit-Zahl ab X um Y erhöhen. Ist Y=0,
                  946  * dann 256 addieren.
65A9: 18          947  ADD      CLC
65AA: 98          948           TYA
65AB: F0 06       949           BEQ  ADD256
65AD: 75 00       950           ADC  0,X
65AF: 95 00       951           STA  0,X
65B1: 90 02       952           BCC  *+4
65B3: F6 01       953  ADD256   INC  1,X
65B5: 60          954           RTS
                  955  PROGLEN  EQU  *-PROGSTART
1580 Bytes
```

Hinweis

Bei dem Programm, das ursprünglich nicht mit dem Big Mac geschrieben worden ist, hat es Tabulierungsprobleme (vgl. z.B. Zeile 152) sowie Konvertierungsprobleme (=< soll => bedeuten) bei der Umwandlung des Quelltextes für die Setzerei gegeben. Das Programm wird jedoch in der vorliegenden Form vom Big Mac fehlerfrei assembliert.

Hex-Monitor

Hilfsprogramm für die Ein- und Ausgabe von Hex-Dumps

von Günter Radestock und Ferdinand Stühlen

Der System-Monitor im Apple II erlaubt die Eingabe von Speicherinhalten nur in einer sehr umständlichen Form. Will man ein Maschinenprogramm als Hex-Zahlen eingeben, so geht man mit
CALL -151
zunächst in den Monitor, gibt die Anfangsadresse, einen Doppelpunkt und darauf die Hex-Zahlen ein, wobei zwischen jedem Byte, also zwei Hex-Ziffern, ein Zwischenraum eingefügt werden muß. Nach spätestens 84 Bytes muß man mit Return abschließen und mit der nachfolgenden Adresse wie oben neu beginnen. Wegen der unformatierten Ausgabe auf dem Bildschirm ist ein Vergleich mit der Vorlage sehr erschwert. Bei der Ausgabe von Speicherstellen auf dem Bildschirm ist man beim System-Monitor auf 8 Bytes pro Zeile festgelegt. Alles in allem eine unbefriedigende Situation.
Das in Applesoft-BASIC geschriebene Programm HEX.MONITOR behebt die aufgeführten Nachteile, es ist mit Ausnahme einiger REM-Zeilen so kurz wie möglich und dennoch verständlich geschrieben.

1. Programmbedienung

Nach dem Start mit
RUN HEX.MONITOR
erscheint die Frage: „Anzahl Bytes pro Zeile 8", wobei der Cursor auf der 8 steht. Gibt man Return ein, so ist der Wert auf 8 Bytes pro Zeile festgelegt. Wünscht man eine andere Byte-Anzahl pro Zeile, so überschreibt man die 8 und gibt dann Return ein. Möchte man beispielsweise einen Hex-Dump eingeben oder vergleichen, der mit 12 Bytes pro Zeile vorliegt, so gibt man 12 ein und hat dann eine zeilenweise übereinstimmende Vergleichsmöglichkeit.
Bei der nächsten Frage „Edit Adresse:$" kann man die Startadresse als Dezimalwert eingeben, indem man das $-Zeichen überschreibt, oder als Hex-Zahl, indem man den Cursor mit der Rechtspfeiltaste eine Stelle nach rechts schiebt und dann den Wert eingibt.
Die letzte Frage „E(ingabe oder D(ump" wird mit E für eine Eingabe und mit D für einen Hex-Dump beantwortet.

Eingabe von Speicherwerten

Hat man die letzte Frage mit E beantwortet, so erscheint am oberen, linken Bildschirmrand die zuvor eingegebene Startadresse in jedem Fall als Hex-Wert mit einem Doppelpunkt. Danach kann die Eingabe der zu speichernden Werte fortlaufend erfolgen. Der Zwischenraum nach jedem Byte, also nach 2 Hex-Ziffern, erfolgt durch das Programm. Ist die Zeile ausgefüllt, so setzt das Programm an den Anfang der nächsten Zeile die entsprechende Speicherzahl mit Doppelpunkt, so daß die Werte fortlaufend ohne Pause eingegeben werden können.
Das Programm nimmt nur die Zeichen von 0-9, A-F und den „-" (Bindestrich) an. Falsche Eingaben werden mit einem Piepston beantwortet, wobei der Cursor auf den ersten Wert des Hex-Ziffernpaares zurückspringt. Statt der Hex-Zahlen ist es auch möglich, ASCII-Zeichen als zweiten Wert des Bytes einzugeben, wenn man als ersten Wert einen Bindestrich eingegeben hat. Das Programm rechnet darauf die Eingabe in den Hex-Code um. Diese Eingabeform ist dann sinnvoll, wenn das Maschinenprogramm Texte enthält.

Editieren

Als Cursor-Funktion können nur die Links- und Rechtspfeiltasten verwendet werden. Der Cursor springt dann von Byte zu Byte. Die Speicherinhalte werden dabei nicht verändert. Führt man den Cursor mit der Linkspfeiltaste auf den Wert der vorhergehenden Zeile, so wird sie als neue Zeile aufgerufen. Darin eingegebene ASCII-Zeichen erscheinen dann umgerechnet als Hex-Werte. Die Eingabe der Hex-Zahlen wird mit Return beendet, worauf die Frage „Neustart (J/N)" erscheint.

Ausgabe von Speicherwerten

Ist die Frage „E(ingabe oder D(ump" mit D beantwortet worden, so muß man auf die Frage „Wieviele Zeilen?" die gewünschte Zeilenzahl eingeben. Die Angabe eines Speicherbereichs ist nicht möglich. Im linken und mittleren Drittel der Ausgabe werden die Adressen und Speicherinhalte und im rechten Drittel die zugehörigen ASCII-Werte dargestellt – die übliche Art, Textinhalte zu erkennen. Die anschließende Frage erlaubt die Ausgabe auf den Drucker.

2. Technisches

Die Programmzeile 140 enthält neben HOME noch CHR$(12) für die Löschung des Bildschirms bei einer 80-Zeichenkarte. Der Drucker-Slot ist in Zeile 840 auf PR#1, der Rücksprung in Zeile 930 auf PR#0 für den 40-Zeichenbildschirm eingestellt. Änderungen auf andere Slot-Werte sind ohne Störung des Programms möglich.

Das Programm selbst hat eine Länge von 1904 Bytes und belegt damit den Speicherplatz von $0800 bis $0F77. Eingaben von Speicherwerten können daher erst ab $0F78 erfolgen, da das Programm sonst überschrieben und zerstört würde. Sollte

ein Maschinenprogramm in dem Bereich des BASIC-Programms liegen, was aber kaum anzunehmen ist, so gibt man es am besten ab Speicherstelle $1000 ein und verschiebt es anschließend. Man kann das eingegebene Programm aber auch mit dem gewählten Speicherbereich auf der Diskette speichern, dann unter der gewünschten Adresse einladen und mit dem neuen Adressenbereich wieder auf der Diskette erneut speichern. Damit umgeht man den Verschiebevorgang im Monitor. Da der Hex-Monitor in Applesoft-BASIC geschrieben ist, kann es im oberen Speicherbereich zu Störungen kommen, weil die Stringvariablen vom oberen Speicherbereich abwärts in ASCII-Form abgelegt werden. Gibt man beispielsweise ein etwas längeres Programm ab $9000 ein, so darf man sich nicht wundern, wenn das mühsam eingegebene Maschinenprogramm mit den Werten von $30 bis $39 und $41 bis $46 (den ASCII-Werten von 0-9 und A-F) überschrieben wird. Man muß in diesem Fall vor Eingabe des Maschinenprogramms mit
HIMEM: 36860
den String-Pool herabsetzen.

Kurzhinweise

1. Zweck:
Applesoft-Programm zur Eingabe von Hex-Dumps
2. Konfiguration:
Apple II+/e/c, 40 Z/Z; DOS 3.3 oder ProDOS
3. Test:
RUN HEX.MONITOR
4. Sammeldisk:
HEX.MONITOR (Applesoft-Programm)

HEX. MONITOR

```
100 REM Eingabe von Hexdumps
110 REM 14.04.85 von Guenter Radestock
        und Ferdinand Stuehlen
120 REM ------------------------------
130 REM
140 HOME : PRINT CHR$ (12);
    "HEX 1.0 (C) 1985 by *** G ***":
    PRINT : PRINT
150 PRINT "Anzahl Bytes pro Zeile:  8";
    CHR$ (8); CHR$ (8);
160 INPUT A$: IF A$ = "" THEN BT = 8:
    PRINT BT: GOTO 180
170 BT = VAL (A$)
180 GOTO 580
190 REM Drucke X mit 2 Stellen in Hex
200 PRINT MID$ ("0123456789ABCDEF",
    INT (X / 16) + 1,1); MID$
    ("0123456789ABCDEF",X -
    16 * INT (X / 16) + 1,1);
210 RETURN
220 REM Eingabe Adresse Hex in X
230 PRINT " $"; CHR$ (8); CHR$ (8);
240 INPUT A$
250 IF LEFT$ (A$,1) = "$" THEN A$ =
    RIGHT$ (A$, LEN (A$) - 1): GOTO 270
260 X = VAL (A$): GOTO 320
270 X = 0
280 FOR I = 1 TO LEN (A$)
290 B$ = MID$ (A$,I,1)
300 X = 16 * X + ASC (B$) - 48 - 7
    * (B$ > "§")
310 NEXT
320 RETURN
330 REM Aufbau einer Zeile Dump
340 PRINT
350 X = INT (ADR / 256): GOSUB 190
360 X = ADR - 256 * X: GOSUB 190
370 PRINT " : ";
380 FOR I = 0 TO BT - 1
390 X = PEEK (ADR + I): GOSUB 190:
    PRINT " ";
400 NEXT
410 REM Cursor auf Byte
420 IF B > BT THEN RETURN
430 FOR I = BT TO B STEP - 1
440 PRINT CHR$ (8); CHR$ (8); CHR$ (8);
450 NEXT
460 RETURN
470 REM Eingabe Byte Hex oder
    ASCII in X, sonst X=-1
480 GET A$: IF A$ < "0" OR (A$ > "9"
    AND A$ < "A") OR A$ > "F" THEN 530
490 PRINT A$;:X = 16 * ( ASC (A$) -
    48 - 7 * (A$ > "'"))
500 GET A$: IF A$ < "0" OR (A$ > "9"
    AND A$ < "A") OR A$ > "F" THEN 520
510 PRINT A$;:X = X + ASC (A$) -
    48 - 7 * (A$ > "'"): RETURN
520 X = - 2: PRINT CHR$ (8);: RETURN
530 IF A$ < > "-" THEN X = - 1: RETURN
540 PRINT A$;: GET A$
550 IF A$ > = " " THEN PRINT A$;:
    GOTO 570
560 PRINT ".";
570 X = ASC (A$): RETURN
580 PRINT "Edit Adresse : ": GOSUB 220
590 ADR = X
600 INPUT "E(ingabe oder D(ump ";A$:
    IF A$ < > "D" AND A$ < > "E"
    THEN 600
610 IF A$ = "D" THEN 830
620 B = 1: GOSUB 330:
    REM Zeile aufbauen
630 GOSUB 470: IF X < 0 THEN 670
640 POKE ADR + B - 1,X:B = B + 1
650 PRINT " ";: IF B > BT THEN ADR
    = ADR + BT: GOTO 620
660 GOTO 630
670 IF A$ < > CHR$ (8) THEN 740
680 IF X = - 2 THEN 630
690 PRINT CHR$ (8); CHR$ (8); CHR$ (8);
700 B = B - 1
710 IF B > 0 THEN 630
720 B = BT:ADR = ADR - BT: GOSUB 330
730 GOTO 630
740 IF A$ = CHR$ (13) THEN 790
750 IF (A$ < > " " AND A$ < >
    CHR$ (21)) OR X < > - 1 THEN
    PRINT CHR$ (7);: GOTO 630
760 B = B + 1: IF B > BT THEN 780
770 X = PEEK (ADR + B - 2): GOSUB 190:
    PRINT " ";: GOTO 630
780 ADR = ADR + BT: GOTO 620
790 PRINT
800 INPUT "Neustart (J/N)";A$:
    IF A$ = "J" THEN RUN
810 IF A$ < > "N" THEN 800
820 END
830 INPUT "Wieviele Zeilen ";L
840 INPUT "Ausgabe auf Drucker (J/N) ";
    DD$: IF DD$ = "J" THEN PRINT:
    PRINT CHR$(4) "PR# 1"
850 FOR J = 1 TO L
860 B = BT + 1: GOSUB 330
870 FOR I = 1 TO BT:
    X = PEEK (ADR + I - 1)
880 IF X > 128 THEN X = X - 128
890 IF X > = 32 AND X < 127
    THEN PRINT CHR$ (X);: GOTO 910
900 PRINT ".";
910 NEXT I:ADR = ADR + BT
920 NEXT J
930 PRINT : PRINT CHR$(4)"PR#0"
940 GOTO 790
```


SUPERQUICK

Ein superschnelles Disketten-Kopierprogramm

von Arne Schäpers, 1985, Programmdiskette mit Anleitung, DM 48,–

Mit SUPERQUICK ist es möglich, Disketten jeden Formats (DOS 3.3, ProDOS, UCSD-Pascal und CP/M) in einer unglaublich kurzen Zeit von nur 29 Sekunden (mit Formatierung) zu kopieren. Bei entsprechender Speichererweiterung kann der gesamte Disketteninhalt eingelesen werden, um mehrere Kopien anzufertigen. Die Zeit für eine Einzelkopie reduziert sich dann auf sage und schreibe 19 Sekunden.

SUPERQUICK erkennt die 64K-Karte (in Slot 3) des Apple IIe und IIc sowie eine 16K-Language-Card in Slot 0 und bezieht diese selbständig als Datenpuffer ein. Darüber hinaus werden die IBS-Karten AP17 in den Ausbaustufen 64K bis 256K automatisch unterstützt und gegebenenfalls als weitere Puffer eingesetzt.

Jetzt mit Spezialprogramm für 160-Spur-Erphi-Laufwerke!

Hüthig Software Service · Postfach 10 28 69 · 6900 Heidelberg 1

Textzeichen

auf dem Double-Hires-Bildschirm

von Steffen Holzinger

Dieser Beitrag wendet sich an jene Leser, die Besitzer eines Apple IIe mit erweiterter 80-Zeichenkarte sind, und beschreibt ein Programm, das es ermöglicht, Textzeichen aus selbsterstellten Zeichensätzen in den hochauflösenden Grafikbildschirm zu schreiben, um damit beispielsweise Grafiken zu beschriften. Das Programm heißt **DHRCG** und ähnelt dem HRCG, das auf der DOS-Tool-Kit-Diskette der Firma Apple zu finden ist. Im Gegensatz zu jenem Programm ist DHRCG jedoch in der Lage, die doppelt-hochauflösende Grafik des Apple IIe zu verwenden. Es erlaubt deshalb bis zu 80 Zeichen pro Grafikzeile.
Da dieser Artikel eine Anwendung der Double-Hires-Grafik zeigt, wird auf die detaillierte Beschreibung des Bildschirmspeichers und der benötigten Softswitches verzichtet. Einzelheiten hierzu liest man am besten bei (1) nach.

1. Eingabe des Programms

1. Zunächst gibt man das Applesoft-Programm **DHRCG.LOADER** ein und speichert es entsprechend ab.
2. Danach gibt man den Hex-Dump des Maschinenprogramms **DHRCG** ein und speichert ihn mit BSAVE DHRCG, A$9000, L$04CF ab. Der Quelltext T.DHRCG ist gesondert gelistet.
3. Schließlich gibt man den deutschen Zeichensatz GERMAN.SET ein und speichert ihn mit BSAVE GERMAN.SET, A$8D00, L$0300 ab.

Sind alle notwendigen Files (DHRCG, DHRCG.LOADER und GERMAN.SET) auf der Diskette vorhanden, so startet man durch
RUN DHRCG.LOADER
das Applesoft-Programm, das DHRCG sowie den primären Zeichensatz (GERMAN.SET) in den Speicher lädt und danach nach weiteren Zeichensätzen fragt. Falls Sie nur den deutschen Zeichensatz benutzen möchten, so geben Sie 0 ein. Falls Sie über weitere Zeichensätze verfügen (z.B. von der DOS-Tool-Kit-Diskette, deren Files natürlich nicht auf der Peeker-Sammeldisk enthalten sind), so geben Sie die gewünschte Anzahl und danach die jeweiligen Namen der Zeichensätze ein. Nach jeder Namenseingabe wird der entsprechende Zeichensatz von Diskette geladen.

2. Betrieb des Programms

Nach dem Start wird auf Double-Hires-Grafik umgeschaltet, der Bildschirm gelöscht und eine Meldung ausgegeben. Danach geht alles normal weiter – mit einigen Ausnahmen:

1. Nach Beendigung einer Eingabezeile wird nicht mehr der Rest der Zeile gelöscht.
2. Applesoft-Programme werden nur bis zur 40. Spalte gelistet, obwohl die Anzeige 80 Zeichen breit ist (gilt nicht für IIe mit neuen ROMs).
3. Ausgabe- und Scrollgeschwindigkeit haben spürbar abgenommen, dies ist aber unvermeidbar.

Von nun an kann man Grafiken und Text mischen und damit beispielsweise Diagramme beschriften. VTAB und HTAB arbeiten wie gewohnt. Ab der 41. Spalte muß man jedoch statt HTAB x den Befehl POKE 36,x-1 verwenden. Zu beachten ist, daß Befehle wie HOME, INVERSE und NORMAL nicht mehr funktionieren (können). Diese Funktionen sind (wie alle Sonderfunktionen des Programms) über Ctrl-Zeichen zu erreichen (s. Tabelle). Ein HOME erreicht man z.B. durch PRINT CHR$(12).
DHRCG bietet einen speziellen OR-Modus, der es ermöglicht, mehrere Zeichen übereinander zu schreiben. Dies ist hilfreich für das Unterstreichen von Wörtern. Der OR-Modus wird mit Ctrl-Q ein- und mit Ctrl-R ausgeschaltet.
Um die Bildschirmfensterbreite zu verändern, darf keinesfalls POKE 33,x verwendet werden, sondern statt dessen PRINT CHR$(2); CHR$(x), wobei x im Bereich von 1-80 liegen muß. Um einen anderen der geladenen Zeichensätze anzusprechen, wird PRINT CHR$(1); CHR$(x) verwendet. Dabei x muß im Bereich von 0 bis 9 liegen.

Vermischte Tips

1. Vermeiden Sie den HGR-Befehl, weil dieser die Bildschirm-Softswitches verändert. Notfalls tippen Sie HGR: POKE –16302, O in einer einzigen Zeile. Statt HGR ist PRINT CHR$(12) vorzuziehen.

2. Vermeiden Sie PR#3. Dies wäre auf alle Fälle "tödlich". Die DHRCG-Routinen werden korrekt mit PRINT CHR$(21) abgeschaltet. Danach können Sie wieder PR#3 eingeben. ESC Ctrl-Q schaltet nicht DHRCG ab!

3. Um sowohl die DHRCG-Routinen als auch eigene Double-Hires-Plot-Routinen verwenden zu können, muß zunächst geprüft werden, inwieweit sich beide Routinen speichermäßig überlappen. Die Peeker-Sammeldisk #22 enthält ein geändertes Startprogramm DHRCG.LOADER und zusätzlich die früheren Ampersand-Routinen. DHRCG.LOADER lädt dann:
   1. DHRCG nach $9000ff.
   2. GERMAN.SET nach $8DOOff.
   3. AMPER.DOUBLE.HIRES nach $6000ff.
   4. AMPER.DOUBLE.HIRES.BAS nach $0800ff.

AMPER.DOUBLE.HIRES ist bereits im Peeker 2/84 beschrieben worden (Quellcode auf Sammeldisk #1). Beide Routinen können gleichzeitig benutzt werden. Ähnlich verfahren Sie mit eigenen Double-Hires-Plot-Routinen.

3. Verschiedenes

Während des Betriebs von DHRCG dürfen MAXFILES, HIMEM und FP nicht benutzt werden, da das Programm HIMEM nach unten verschiebt, um Platz für sich selbst und für die Zeichensätze zu schaffen. Aus diesem Grund läuft es auch nicht unter ProDOS.
Die augenblickliche horizontale Cursorposition ist durch PEEK (36) zu erfahren und liegt im Bereich von 0-79.
Durch PRINT CHR$(21) kann das Programm vollkommen von der Ein- und Ausgabe abgetrennt werden. Außerdem wird auf den 40-Zeichen-Textbildschirm zurückgeschaltet.
Alle Zeichensätze von der DOS-Tool-Kit-Diskette können verwendet werden, wobei zu beachten ist, daß manche dieser Zeichensätze unter DHRCG „etwas merkwürdig" aussehen, weil das Setzen von Bit 7 eines Bytes im Bildschirmspeicher nicht mehr eine Verschiebung um einen halben Punkt bedeutet.

4. Erstellen von Zeichensätzen

Ein Zeichensatz besteht aus 768 ($300) Bytes und enthält die Zeichen mit den ASCII-Codes von 32-127, wobei jedes Zeichen 8 Bytes belegt. Am Anfang des Zeichensatzes stehen also 8 Bytes, die CHR$(32) = Space definieren, danach 8 Bytes für CHR$(33) = Ausrufezeichen usw. bis zu den letzten 8 Bytes für CHR$(127) = Del. Der Aufbau der Bytes ergibt sich aus der untenstehenden Zeichnung. Jeder gesetzte Punkt innerhalb einer Zeile der Matrix entspricht einem gesetzten Bit in dem jeweiligen Byte. Bit 7 ist nicht relevant, denn das Setzen des höchsten Bits bewirkt im Gegensatz zur normal-hochauflösenden Grafik keine Verschiebung um einen halben Punkt.
Mit der Definition des Zeichensatzes kann man an jeder beliebigen Adresse im Speicher beginnen (z.B. $4000). So ergeben sich jeweils die benötigten 8 Bytes pro Zeichen. Ist die mühevolle Definition für jedes der 96 Zeichen abgeschlossen, so kann man den Zeichensatz mit BSAVE NAMEXYZ, A$4000, L$300 abspeichern. Besitzt man die DOS-Tool-Kit-Disk, so kann man eigene Zeichensätze mit ANIMATRIX erstellen, was wesentlich einfacher als der Entwurf „per Hand" ist.

Literatur

(1) Karl-Walter Bott: Wie man die Grafik verdoppelt, Teil 2: Double Hires; Peeker 2/1984
(2) Apple Computer: Extended 80-Column Text Card Supplement, Cupertino 1982
(3) Apple Computer: DOS Tool-Kit, 1979

Hinweis zur Sammeldisk

Dateien:
DHRCG.LOADER
T.DHRCG
DHRCG
GERMAN.SET
Starten:
RUN DHRCG.LOADER
unter DOS 3.3 auf IIe.

Tabelle der DHRCG-Ctrl-Steuerzeichen

Ctrl	ASCII		Bedeutung
A	1	$81	Schaltet Zeichensatz ein
B	2	$82	Setzt neue Bildschirmbreite
C	3	$83	Initialisiert alle Einstellungen
G	7	$87	Gibt Standardpiepser aus
H	8	$88	Cursor einen Schritt zurück
J	1Ø	$8A	Cursor eine Zeile nach unten
K	11	$8B	ClrEOP
L	12	$8C	Bildschirm löschen
M	13	$8D	Return ausführen
N	14	$8E	Schaltet NORMAL ein
O	15	$8F	Schaltet INVERSE ein
Q	17	$91	Schaltet OR-Modus ein
R	18	$92	Schaltet OR-Modus aus
U	21	$95	Schaltet DHRCG aus
W	23	$97	Scrollt Bildschirm um eine Zeile
Y	25	$99	Setzt Cursor in linke obere Ecke
Z	26	$9A	Löscht gesamte Cursorzeile
Ö	28	$9C	Bewegt Cursor einen Schritt weiter
Ü	29	$9D	ClrEOL

Tabelle zum Zeichensatzaufbau

rel. Byte	Ø123456 <- Bits	
Ø	..***..	$1C
1	.*...*.	$22
2	.*.....	$Ø2
3	..***..	$1C
4	.....*.	$2Ø
5	.*...*.	$22
6	..***..	$1C
7	.....	$ØØ

Gelöschtes Bit = .
Gesetztes Bit = *

Obige Tabelle beschreibt die Definition des Buchstabens „S" innerhalb eines DHRCG-Zeichensatzes.

DHRCG.LOADER
Applesoft-Startprogramm

```
1ØØ PRINT CHR$ (21)
11Ø PRINT CHR$ (4);
    "BLOAD DHRCG,A$9ØØØ"
12Ø PRINT CHR$ (4);
    "BLOAD GERMAN.SET,A$8DØØ"
13Ø TEXT : HOME
14Ø INPUT "Wieviele Zeichensätze
          (Ø-9) --> ";A$
15Ø N = VAL (A$): IF N < > INT (N)
    OR N < Ø OR N > 9
    THEN PRINT CHR$ (7);: GOTO 14Ø
16Ø AD = 36Ø96: REM $8DØØ
17Ø TBL = 36873: REM $9ØØ9
18Ø HM = AD - N * 768: HIMEM: HM
19Ø POKE TBL,Ø: POKE TBL + 1,141
2ØØ IF N = Ø THEN 29Ø
21Ø HOME
22Ø FOR I = 1 TO N
23Ø PRINT "ZeicheensatGz Nr. ";I;:
    INPUT " --> ";A$
24Ø AD = AD - 768
25Ø PRINT CHR$ (4);"BLOAD";A$;",A";AD
26Ø AH% = AD / 256:AL% = AD - AH% * 256
27Ø POKE TBL + I * 2,AL%:
    POKE TBL + 1 + I * 2,AH%
28Ø NEXT
29Ø CALL 36864: REM $9ØØØ
3ØØ END
```

T.DHRCG
Big-Mac-Quelltext

```
1    ********************************
4    * Double-Hires Zeichengenerator *
5    * ----------------------------- *
8    *          Version 1.3          *
11   *   Steffen Holzinger 1985-86   *
14   ********************************
15
16   WNDLFT   EQU  $2Ø
17   WNDTOP   EQU  $22
18   WNDBTM   EQU  $23
19   CH       EQU  $24
2Ø   CV       EQU  $25
21   GBASL    EQU  $26
22   GBASH    EQU  $27
23   BASL     EQU  $28
24   BASH     EQU  $29
25   BAS2L    EQU  $2A
26   BAS2H    EQU  $2B
27   INVFLG   EQU  $32
28   CSWL     EQU  $36
29   CSWH     EQU  $37
3Ø   KSWL     EQU  $38
31   KSWH     EQU  $39
32   RNDL     EQU  $4E
33   RNDH     EQU  $4F
34   SETAD3   EQU  $FA
35   GBAS2L   EQU  $FC
36   GBAS2H   EQU  $FD
37   CONCT1   EQU  $Ø3EA
38   MODE     EQU  $Ø4FB
39   KBD      EQU  $CØØØ
4Ø   CLR8ØCOL EQU  $CØØØ
41   SET8ØCOL EQU  $CØØ1
42   CLRALTZP EQU  $CØØ8
43   SETALTZP EQU  $CØØ9
44   CLR8ØVID EQU  $CØØC
45   SET8ØVID EQU  $CØØD
46   CLRALTCH EQU  $CØØE
47   KBDSTRB  EQU  $CØ1Ø
48   CLRTEXT  EQU  $CØ5Ø
49   CLRMIXED EQU  $CØ52
5Ø   CLRPAGE2 EQU  $CØ54
51   SETPAGE2 EQU  $CØ55
52   SETHIRES EQU  $CØ57
53   CLRAN3   EQU  $CØ5E
54   INIT     EQU  $FB2F
55   SYSBYT   EQU  $FBB3
56   BELL1    EQU  $FBD9
57   CLRSCR   EQU  $FC58
58   COUT     EQU  $FDED
59   SETKBD   EQU  $FE89
6Ø   SETVID   EQU  $FE93
61
62   ********************************
63            ORG  $9ØØØ
64   ********************************
65
66   *
67   * Vektoren
68   *
69   START    JMP  COLD
7Ø            JMP  WARM
71   CONNECT  JMP  CONCT1
72   *
73   * Zeichensatz-Anfangsadressen
74   *
75   SETØADR  DA   Ø
```

```
76   SET1ADR  DA   Ø
77   SET2ADR  DA   Ø
78   SET3ADR  DA   Ø
79   SET4ADR  DA   Ø
8Ø   SET5ADR  DA   Ø
81   SET6ADR  DA   Ø
82   SET7ADR  DA   Ø
83   SET8ADR  DA   Ø
84   SET9ADR  DA   Ø
85   *
86   * Cursor-Blinkgeschwindigkeit
87   *
88   BLOFF    DFB  Ø
89
9Ø   ********************************
91   *                              *
92   * Initialisierung              *
93   *                              *
94   ********************************
95
96   WARM     JSR  CARDOFF
97            JSR  CHKSYS
98            JSR  SETSCR
99            JSR  SETIO
1ØØ           JSR  CONNECT
1Ø1           JMP  NEWPAR
1Ø2
1Ø3  COLD     JSR  WARM
1Ø4           JMP  HEADLN
1Ø5
1Ø6  ********************************
1Ø7
1Ø8  *
1Ø9  * 8Ø-Zeichen ausschalten
11Ø  *
111  CARDOFF  LDA  #$FF
112           STA  MODE
113           STA  CLR8ØVID
114           STA  CLRALTCH
115           JSR  SETVID
116           JMP  SETKBD
117  *
118  * Softswitches für Double-Hires
119  *
12Ø  SETSCR   LDA  SETHIRES
121           LDA  CLRPAGE2
122           LDA  CLRMIXED
123           STA  SET8ØVID
124           STA  SET8ØCOL
125           LDA  CLRAN3
126           LDA  CLRTEXT
127           RTS
128  *
129  * $36-$39 setzen
13Ø  *
131  SETIO    LDY  #OUT
132           LDA  #OUT/256
133           STY  CSWL
134           STA  CSWH
135           LDY  #IN
136           LDA  #IN/256
137           STY  KSWL
138           STA  KSWH
139           RTS
14Ø  *
141  * Variablen initialisieren
142  *
143  NEWPAR   LDY  #$ØØ
144           STY  SETNUM
145           JSR  NRM
146           JSR  CLROR
147           LDA  #$5Ø
148           STA  WNDWDTH
149           LSR  SETMODE
15Ø           LSR  WNDMODE
151           LDA  #$3F
152           STA  BLOFF
153           RTS
154  *
155  * Auf 128K-Apple IIe prüfen
156  *
157  CHKSYS   PHP
158           SEI
159           LDA  SYSBYT
16Ø           CMP  #$Ø6
161           BNE  CHKSYS1
162           LDA  #$A5
163           STA  SETALTZP
164           STA  $1ØØ
165           EOR  $1ØØ
166           STA  CLRALTZP
167           BNE  CHKSYS1
168           PLP
169           RTS
17Ø  *
171  * falsches Apple-System
172  *
173  CHKSYS1  JSR  CLRSCR
174           LDY  #$ØØ
175  CHKSYS2  LDA  TXØ,Y
176           JSR  COUT
177           INY
178           CPY  #TX1-TXØ
179           BCC  CHKSYS2
18Ø           PLP
181           PLA
182           PLA
183           PLA
184           PLA
185           JMP  CONNECT
186  *
187  * Titelzeile ausgeben
188  *
189  HEADLN   LDY  #$ØØ
19Ø  HEADLN1  LDA  TX1,Y
191           JSR  COUT
192           INY
193           CPY  #TX2-TX1
194           BCC  HEADLN1
195           RTS
196
197  ********************************
198  *                              *
199  * Berechnungen                 *
2ØØ  *                              *
2Ø1  ********************************
2Ø2
2Ø3  *
2Ø4  * Zeichensatz-Adresse aus
2Ø5  * Tabelle holen
2Ø6  *
2Ø7  CLCSET   PHA
2Ø8           LDA  SETNUM
2Ø9           ASL
21Ø           TAY
211           LDA  SETØADR,Y
212           STA  SETAD1
213           LDA  SETØADR+1,Y
214           STA  SETAD1+1
215           PLA
216           RTS
217  *
218  * Adresse eines Zeichens be-
219  * rechnen
22Ø  * (Adr=SetAdr+8*(Asc-32))
221  *
222  CLCCHR   LDX  #$ØØ
223           STX  SETAD2+1
224           PHA
225           SEC
226           SBC  #$2Ø
227           LDX  #$Ø3
228  CLCCH1   ASL
229           ROL  SETAD2+1
23Ø           DEX
231           BNE  CLCCH1
232           ADC  SETAD1
233           STA  SETAD2
234           LDA  SETAD1+1
235           ADC  SETAD2+1
236           STA  SETAD2+1
237           PLA
238
239  AD2AD3   LDX  SETAD2
24Ø           STX  SETAD3
241           LDX  SETAD2+1
242           STX  SETAD3+1
243           RTS
244
245  ********************************
246  *                              *
247  * Bildschirmadressierung       *
248  *                              *
249  ********************************
25Ø
251  *
252  * Je nach Cursorposition Page
253  * 1 bzw. Page 1x einschalten
254  * und Offset speichern
255  *
256  SETCUR   JSR  VTAB
257  HTAB     LDA  CH
258           LSR
259           STA  OFFSET
26Ø           TAY
261           STY  SETPAGE2
262           BCC  HTAB1
263           STY  CLRPAGE2
264  HTAB1    RTS
265  *
266  * Zeilenadresse berechnen
267  *
268  VTAB     LDA  CV
269  VTABZ    JSR  BASCALC
27Ø           ADC  WNDLFT
271           STA  BASL
272           STA  GBASL
273           RTS
274  *
275  * BASCALC-Routine ähnlich
276  * Monitor-Routine
277  *
278  BASCALC  PHA
279           LSR
28Ø           AND  #$Ø3
281           PHA
282           ORA  #$2Ø
283           STA  GBASH
284           PLA
285           ORA  #$Ø4
286           STA  BASH
287           PLA
288           AND  #$18
289           BCC  BASCLC2
29Ø           ADC  #$7F
291  BASCLC2  STA  GBASL
292           ASL
293           ASL
294           ORA  GBASL
295           STA  GBASL
296           RTS
297  *
298  * Zeichen aus Bildschirmspeicher
299  * der 8Ø-Zeichen-Karte holen
3ØØ  *
3Ø1  PICK     STA  SET8ØCOL
3Ø2           JSR  HTAB
3Ø3           LDA  (BASL),Y
3Ø4           STA  CLRPAGE2
3Ø5           STA  CLR8ØCOL
3Ø6           RTS
3Ø7  *
3Ø8  * Zeichen in Bildschirmspeicher
3Ø9  * der 8Ø-Zeichen-Karte...
31Ø  *
311  STORE    PHP
312           JSR  CLCCHR
313           JSR  SETCUR
314           LDA  AREG
315           ORA  #$8Ø
316           BIT  INVFLG
317           BMI  STORE1
318           EOR  #$8Ø
319  STORE1   STA  (BASL),Y
32Ø           PLP
321  *
322  * ...und in das Grafik-RAM
323  *
324  SCRWR    ROR  TEMP
325           JSR  AD2AD3
326           LDA  GBASH
327           PHA
328           LDY  #$ØØ
329  SCRWR1   LDA  (SETAD3),Y
33Ø           EOR  INV2FLG
331           STY  YSAV
332           LDY  OFFSET
333           BIT  TEMP
334           BPL  SCRWR2
335           ORA  (GBASL),Y
336  SCRWR2   STA  (GBASL),Y
337           CLC
338           LDA  #$Ø4
339           ADC  GBASH
34Ø           STA  GBASH
341           LDY  YSAV
342           INY
343           CPY  #$Ø8
344           BCC  SCRWR1
345           PLA
346           STA  GBASH
347           RTS
348  *
349  * Zeichen in der Grafik in-
35Ø  * vertieren (für Cursor)
351  *
352  INVERT   LDA  GBASH
353           PHA
354           TYA
355           PHA
356           STA  SET8ØCOL
357           JSR  SETCUR
```

```
358            LDX  #$08
359  INV1      LDA  (GBASL),Y
360            EOR  #$7F
361            STA  (GBASL),Y
362            CLC
363            LDA  #$04
364            ADC  GBASH
365            STA  GBASH
366            DEX
367            BNE  INV1
368            STA  CLRPAGE2
369            PLA
370            TAY
371            PLA
372            STA  GBASH
373            RTS
374
375  ********************************
376  *                              *
377  * Eingabe                      *
378  *                              *
379  ********************************
380
381  IN        JSR  REGSAVE
382            STA  (BASL),Y
383            JSR  CLCSET
384            PHP
385            SEI
386            JSR  RDCHAR
387            CMP  #$95        ;->
388            BNE  IN2
389            JSR  PICK
390  IN2       STA  AREG
391            JSR  EXIT
392            PLP
393            RTS
394  *
395  * Tastatureingabe mit Blink-
396  * Cursor
397  *
398  RDKEY     LDA  #$00
399            PHA
400            JSR  INVERT
401  RDKEY1    INC  RNDL
402            BNE  RDKEY2
403            INC  RNDH
404  RDKEY2    LDA  RNDH
405            AND  BLOFF
406            BNE  RDKEY3
407            PLA
408            CLC
409            ADC  #$01
410            PHA
411            JSR  INVERT
412            INC  RNDH
413  RDKEY3    LDY  KBD
414            BPL  RDKEY1
415            PLA
416            LSR
417            BCS  RDKEY4
418            JSR  INVERT
419  RDKEY4    TYA
420            STA  KBDSTRB
421            RTS
422  *
423  * Eingabe mit Esc-Sequenzen
424  *
425  ESC       JSR  ESC1
426  RDCHAR    JSR  RDKEY
427            CMP  #$9B        ;ESC
428            BEQ  ESC
429            RTS
430  *
431  * Esc-Kommando ausführen
432  *
433  ESC1      JSR  RDKEY
434            CMP  #$E0
435            BCC  RDCH1
436            SBC  #$20
437  RDCH1     LDY  #$0B
438  RDCH2     DEY
439            BMI  RDCH3
440            CMP  ESCTBL,Y
441            BNE  RDCH2
442            CPY  #$08
443            ROR  ESCMODE
444            JSR  ESCSUBR
445            BIT  ESCMODE
446            BPL  ESC1
447            LSR  ESCMODE
448  RDCH3     RTS
449  *
450  * Aufruf der Esc-Routinen
451  *
```

```
452  ESCSUBR   TYA
453            ASL
454            TAY
455            LDA  ESCADR+1,Y
456            PHA
457            LDA  ESCADR,Y
458            PHA
459            RTS
460
461  ESCTBL    ASC  "I"
462            DFB  $8B         ;↑K
463            ASC  "J"
464            DFB  $88         ;↑H
465            ASC  "K"
466            DFB  $95         ;↑U
467            ASC  "M"
468            DFB  $8A         ;↑J
469            ASC  "§"
470            ASC  "E"
471            ASC  "F"
472  ESCADR    DA   CURUP-1
473            DA   CURUP-1
474            DA   BCKSPC-1
475            DA   BCKSPC-1
476            DA   ADV-1
477            DA   ADV-1
478            DA   LF-1
479            DA   LF-1
480            DA   CLS-1
481            DA   CLREOL-1
482            DA   CLREOP-1
483
484  ********************************
485  *                              *
486  * Zeichenausgabe               *
487  *                              *
488  ********************************
489
490  OUT       JSR  REGSAVE
491            PHP
492            SEI
493            STA  SET80COL
494            JSR  CLCSET
495            LDA  AREG
496            AND  #$7F
497            JSR  WAIT
498            JSR  OUT1
499            JSR  EXIT
500            PLP
501            RTS
502
503  ********************************
504  *                              *
505  * I/O-Subroutinen              *
506  *                              *
507  ********************************
508
509  REGSAVE   STA  AREG
510            STX  XREG
511            STY  YREG
512            RTS
513
514  EXIT      LDY  YREG
515            LDX  XREG
516            LDA  AREG
517            STA  CLRPAGE2
518            STA  CLR80COL
519            RTS
520
521  WAIT      CMP  #$0D        ;CR
522            BNE  WAIT2
523            LDY  KBD
524            BPL  WAIT2
525            CPY  #$93
526            BNE  WAIT2
527            STA  KBDSTRB
528  WAIT1     BIT  KBD
529            BPL  WAIT1
530            STA  KBDSTRB
531  WAIT2     RTS
532
533  ********************************
534  *                              *
535  * Ausgabe-Subroutinen          *
536  *                              *
537  ********************************
538
539  *
540  * Unterscheidung zwischen Text-
541  * und Controlzeichen
542  *
543  OUT1      BIT  WNDMODE
544            BPL  OUT2
545            LSR  WNDMODE
```

```
546            STA  WNDWDTH
547            RTS
548  OUT2      CMP  #$20
549            BCS  PRINT
550            ASL
551            TAY
552            LDA  ADRTBL+1,Y
553            PHA
554            LDA  ADRTBL,Y
555            PHA
556            RTS
557  *
558  * Textzeichen ausgeben
559  *
560  PRINT     PHA
561            BIT  SETMODE
562            BPL  PRT1
563            LSR  SETMODE
564            EOR  #$30
565            STA  SETNUM
566            PLA
567            RTS
568  PRT1      JSR  CLCCHR
569            LDA  ORMODE
570            ROL
571            PLA
572            JSR  STORE
573            BCS  ADV
574
575  ********************************
576
577  NEWWND    SEC
578            ROR  WNDMODE
579            RTS
580
581  NEWSET    SEC
582            ROR  SETMODE
583  NOP       RTS
584
585  BELL      LDA  #$87
586            JMP  BELL1
587  *
588  * Cursor-Vorwärtsbewegung
589  *
590  ADV       INC  CH
591            LDA  CH
592            CMP  WNDWDTH
593            BCS  CR
594            RTS
595  *
596  * Cursor-Rückwärtsbewegung
597  *
598  BCKSPC    DEC  CH
599            BPL  NOP
600            LDA  WNDWDTH
601            STA  CH
602            DEC  CH
603  *
604  * Cursor nach oben
605  *
606  CURUP     LDA  WNDTOP
607            CMP  CV
608            BCS  CURUP1
609            DEC  CV
610  CURUP1    RTS
611  *
612  * Löschen: Cursor bis Bild-
613  *          schirmende
614  *
615  CLREOP    LDY  CH
616            LDA  CV
617  CLEOP1    PHA
618            JSR  VTABZ
619            JSR  CLEOL1
620            LDY  #$00
621            PLA
622            ADC  #$00
623            CMP  WNDBTM
624            BCC  CLEOP1
625            RTS
626  *
627  * Cursor in Zeile 0, Spalte 0
628  *
629  HOME      LDA  WNDTOP
630            STA  CV
631            JSR  VTAB
632            LDY  #$00
633            STY  CH
634            RTS
635  *
636  * Bildschirm löschen
637  *
638  CLS       JSR  HOME
639            LDA  WNDTOP
```

```
640            JMP  CLEOP1
641  *
642  * Carriage Return
643  *
644  CR        LDA  #$ØØ
645            STA  CH
646  *
647  * Line Feed
648  *
649  LF        INC  CV
65Ø            LDA  CV
651            CMP  WNDBTM
652            BCS  LF1
653            JMP  VTABZ
654  LF1       DEC  CV
655  *
656  * Scrolling: Grafik- und Text-
657  *            bildschirm
658  *
659  SCRL      LDA  WNDTOP
66Ø            PHA
661            JSR  VTABZ
662  SCRL1     LDA  GBASL
663            STA  GBAS2L
664            LDA  GBASH
665            AND  #$E3
666            STA  GBAS2H
667            LDA  BASL
668            STA  BAS2L
669            LDA  BASH
67Ø            STA  BAS2H
671            PLA
672            CLC
673            ADC  #$Ø1
674            CMP  WNDBTM
675            BCS  SCRL4
676            PHA
677            JSR  VTABZ
678            LDX  #$Ø7
679  SCRL2     LDA  WNDWDTH
68Ø            LSR
681            TAY
682            DEY
683  SCRL3     STA  SETPAGE2
684            LDA  (GBASL),Y
685            STA  (GBAS2L),Y
686            LDA  (BASL),Y
687            STA  (BAS2L),Y
688            STA  CLRPAGE2
689            LDA  (GBASL),Y
69Ø            STA  (GBAS2L),Y
691            LDA  (BASL),Y
692            STA  (BAS2L),Y
693            DEY
694            BPL  SCRL3
695            DEX
696            BMI  SCRL1
697            CLC
698            LDA  GBASH
699            ADC  #$Ø4
7ØØ            STA  GBASH
7Ø1            LDA  GBAS2H
7Ø2            ADC  #$Ø4
7Ø3            STA  GBAS2H
7Ø4            BNE  SCRL2
7Ø5  SCRL4     JSR  CLRLN
7Ø6            JMP  VTAB
7Ø7  *
7Ø8  * Cursorzeile löschen
7Ø9  *
71Ø  CLRLN     LDY  #$ØØ
711            DFB  $2C          ;BIT
712  *
713  * Löschen: Cursor bis Zeilen-
714  *            ende
715  *
716  CLREOL    LDY  CH
717            JSR  VTAB
718  CLEOL1    LDA  GBASH
719            STA  GBAS2H
72Ø  CLEOL2    STY  YSAV2
721            TYA
722            LSR
723            TAY
724            STA  SETPAGE2
725            BCC  CLEOL3
726            STA  CLRPAGE2
727  CLEOL3    LDX  #$Ø8
728  CLEOL4    LDA  #$ØØ
729            EOR  INV2FLG
73Ø            STA  (GBASL),Y
```

```
731            CLC
732            LDA  GBASH
733            ADC  #$Ø4
734            STA  GBASH
735            DEX
736            BNE  CLEOL4
737            LDA  #$AØ
738            BIT  INVFLG
739            BMI  CLEOL5
74Ø            EOR  #$8Ø
741  CLEOL5    STA  (BASL),Y
742            STA  CLRPAGE2
743            LDA  GBAS2H
744            STA  GBASH
745            LDY  YSAV2
746            INY
747            CPY  WNDWDTH
748            BCC  CLEOL2
749            RTS
75Ø  *
751  * OR-Modus setzen/löschen
752  *
753  SETOR     SEC
754            DFB  $9Ø          ;BCC
755  CLROR     CLC
756            ROR  ORMODE
757            RTS
758  *
759  * NORMAL/INVERSE setzen
76Ø  *
761  NRM       LDY  #$ØØ
762            STY  INV2FLG
763            DEY
764            STY  INVFLG
765            RTS
766  INV       LDA  #$7F
767            STA  INV2FLG
768            LDA  #$3F
769            STA  INVFLG
77Ø            RTS
771  *
772  * Abschalten der Karte
773  *
774  BYE       JSR  INIT
775            JSR  CARDOFF
776            JSR  CLRSCR
777            JMP  CONNECT
778
779  ********************************
78Ø  *                              *
781  * Variablen und Texte          *
782  *                              *
783  ********************************
784
785  ADRTBL    DA   NOP-1        ;$
786            DA   NEWSET-1     ;A
787            DA   NEWWND-1     ;B
788            DA   NOP-1        ;C
789            DA   NEWPAR-1     ;D
79Ø            DA   NOP-1        ;E
791            DA   NOP-1        ;F
792            DA   BELL-1       ;G
793            DA   BCKSPC-1     ;H
794            DA   NOP-1        ;I
795            DA   LF-1         ;J
796            DA   CLREOP-1     ;K
797            DA   CLS-1        ;L
798            DA   CR-1         ;M
799            DA   NRM-1        ;N
8ØØ            DA   INV-1        ;O
8Ø1            DA   NOP-1        ;P
8Ø2            DA   SETOR-1      ;Q
8Ø3            DA   CLROR-1      ;R
8Ø4            DA   NOP-1        ;S
8Ø5            DA   NOP-1        ;T
8Ø6            DA   BYE-1        ;U
8Ø7            DA   NOP-1        ;V
8Ø8            DA   SCRL-1       ;W
8Ø9            DA   NOP-1        ;X
81Ø            DA   HOME-1       ;Y
811            DA   CLRLN-1      ;Z
812            DA   NOP-1        ;Ä
813            DA   ADV-1        ;Ö
814            DA   CLREOL-1     ;Ü
815            DA   NOP-1        ;↑
816            DA   NOP-1        ;_
817
818  ********************************
819
82Ø  AREG      DFB  Ø
821  XREG      DFB  Ø
```

```
822  YREG      DFB  Ø
823
824  SETAD1    DA   Ø
825  SETAD2    DA   Ø
826
827  YSAV      DFB  Ø
828  YSAV2     DFB  Ø
829
83Ø  SETNUM    DFB  Ø
831  WNDWDTH   DFB  Ø
832  ORMODE    DFB  Ø
833  INV2FLG   DFB  Ø
834  ESCMODE   DFB  Ø
835  SETMODE   DFB  Ø
836  WNDMODE   DFB  Ø
837
838  TEMP      DFB  Ø
839  OFFSET    DFB  Ø
84Ø
841  ********************************
842
843  TXØ       ASC  "***  64K-KARTE "
844            ASC  "FEHLT  ***"
845            DFB  $87,$87
846            DFB  $8D,$8D
847  TX1       DFB  $8C
848            ASC  "Double-Hires "
849            ASC  "Zeichengenerator"
85Ø            DFB  $8D
851            ASC  "Steffen Holzinger "
852            ASC  "1985-86"
853            DFB  $8D
854  TX2       EQU  *
855  *         1227 Bytes
```


Softbreaker 1.0

Eine softwaremäßige Interrupt-Utility für die Apple IIe 64K-Karte

von U. Stiehl

1984, Diskette und Manual, DM 20,–
ISBN 3-7785-1022-3

Produkt läuft aus. Ab 1. 6. 86 Diskette mit Quellcode für nur noch DM 20,–

Softbreaker ist ein Assemblerprogramm, mit dessen Hilfe Programme, die sich von der 64K-Karte (= Extended 80 Column Card für den Apple IIe) starten lassen, unterbrochen, gespeichert, geladen und exakt an der Stelle der Unterbrechung fortgeführt werden können. Dadurch ist es auch möglich, Sicherungskopien von sogenannten kopiergeschützten Programmen herzustellen.

Mit Softbreaker unterbrochene Programme werden komplett, d. h. die ganzen 64K einschließlich Language Card, in nur ca. 11 Sekunden auf einer formatierten Diskette gesichert.

Gerätevoraussetzung: Apple IIe mit 64K-Karte, nicht IIc, nicht neue ROMs

Hüthig Software Service,
Postfach 10 28 69, D-6900 Heidelberg

DHRCG
Hex-Dump

```
$9000: 4C 30 90 4C 1E 90 4C EA
$9008: 03 00 00 00 00 00 00 00
$9010: 00 00 00 00 00 00 00 00
$9018: 00 00 00 00 00 00 20 36
$9020: 90 20 8A 90 20 47 90 20
$9028: 5D 90 20 06 90 4C 6E 90
$9030: 20 1E 90 4C BD 90 A9 FF
$9038: 8D FB 04 8D 0C C0 8D 0E
$9040: C0 20 93 FE 4C 89 FE AD
$9048: 57 C0 AD 54 C0 AD 52 C0
$9050: 8D 0D C0 8D 01 C0 AD 5E
$9058: C0 AD 50 C0 60 A0 6B A9
$9060: 92 84 36 85 37 A0 C5 A9
$9068: 91 84 38 85 39 60 A0 00
$9070: 8C 6C 94 20 04 94 20 FF
$9078: 93 A9 50 8D 6D 94 4E 71
$9080: 94 4E 72 94 A9 3F 8D 1D
$9088: 90 60 08 78 AD B3 FB C9
$9090: 06 D0 12 A9 A5 8D 09 C0
$9098: 8D 00 01 4D 00 01 8D 08
$90A0: C0 D0 02 28 60 20 58 FC
$90A8: A0 00 B9 75 94 20 ED FD
$90B0: C8 C0 1D 90 F5 28 68 68
$90B8: 68 68 4C 06 90 A0 00 B9
$90C0: 92 94 20 ED FD C8 C0 39
$90C8: 90 F5 60 48 AD 6C 94 0A
$90D0: A8 B9 09 90 8D 66 94 B9
$90D8: 0A 90 8D 67 94 68 60 A2
$90E0: 00 8E 69 94 48 38 E9 20
$90E8: A2 03 0A 2E 69 94 CA D0
$90F0: F9 6D 66 94 8D 68 94 AD
$90F8: 67 94 6D 69 94 8D 69 94
$9100: 68 AE 68 94 86 FA AE 69
$9108: 94 86 FB 60 20 1F 91 A5
$9110: 24 4A 8D 74 94 A8 8C 55
$9118: C0 90 03 8C 54 C0 60 A5
$9120: 25 20 2B 91 65 20 85 28
$9128: 85 26 60 48 4A 29 03 48
$9130: 09 20 85 27 68 09 04 85
$9138: 29 68 29 18 90 02 69 7F
$9140: 85 26 0A 0A 05 26 85 26
$9148: 60 8D 01 C0 20 0F 91 B1
$9150: 28 8D 54 C0 8D 00 C0 60
$9158: 08 20 DF 90 20 0C 91 AD
$9160: 63 94 09 80 24 32 30 02
$9168: 49 80 91 28 28 6E 73 94
$9170: 20 01 91 A5 27 48 A0 00
$9178: B1 FA 4D 6F 94 8C 6A 94
$9180: AC 74 94 2C 73 94 10 02
$9188: 11 26 91 26 18 A9 04 65
$9190: 27 85 27 AC 6A 94 C8 C0
$9198: 08 90 DD 68 85 27 60 A5
$91A0: 27 48 98 48 8D 01 C0 20
$91A8: 0C 91 A2 08 B1 26 49 7F
$91B0: 91 26 18 A9 04 65 27 85
$91B8: 27 CA D0 F0 8D 54 C0 68
$91C0: A8 68 85 27 60 20 86 92
$91C8: 91 28 20 CB 90 08 78 20
$91D0: 12 92 C9 95 D0 03 20 49
$91D8: 91 8D 63 94 20 90 92 28
$91E0: 60 A9 00 48 20 9F 91 E6
$91E8: 4E D0 02 E6 4F A5 4F 2D
$91F0: 1D 90 D0 0A 68 18 69 01
$91F8: 48 20 9F 91 E6 4F AC 00
$9200: C0 10 E4 68 4A B0 03 20
$9208: 9F 91 98 8D 10 C0 60 20
$9210: 1A 92 20 E1 91 C9 9B F0
$9218: F6 60 20 E1 91 C9 E0 90
$9220: 02 E9 20 A0 0B 88 30 15
$9228: D9 4A 92 D0 F8 C0 08 6E
$9230: 70 94 20 3E 92 2C 70 94
$9238: 10 E0 4E 70 94 60 98 0A
$9240: A8 B9 56 92 48 B9 55 92
$9248: 48 60 C9 8B CA 88 CB 95
$9250: CD 8A C0 C5 C6 14 93 14
$9258: 93 09 93 09 93 FF 92 FF
$9260: 92 4A 93 4A 93 3E 93 B7
$9268: 93 1D 93 20 86 92 08 78
$9270: 8D 01 C0 20 CB 90 AD 63
$9278: 94 29 7F 20 A0 92 20 B9
$9280: 92 20 90 92 28 60 8D 63
$9288: 94 8E 64 94 8C 65 94 60
$9290: AC 65 94 AE 64 94 AD 63
$9298: 94 8D 54 C0 8D 00 C0 60
$92A0: C9 0D D0 14 AC 00 C0 10
$92A8: 0F C0 93 D0 0B 8D 10 C0
$92B0: 2C 00 C0 10 FB 8D 10 C0
$92B8: 60 2C 72 94 10 07 4E 72
$92C0: 94 8D 6D 94 60 C9 20 B0
$92C8: 0B 0A A8 B9 24 94 48 B9
$92D0: 23 94 48 60 48 2C 71 94
$92D8: 10 0A 4E 71 94 49 30 8D
$92E0: 6C 94 68 60 20 DF 90 AD
$92E8: 6E 94 2A 68 20 58 91 B0
$92F0: 0F 38 6E 72 94 60 38 6E
$92F8: 71 94 60 A9 87 4C D9 FB
$9300: E6 24 A5 24 CD 6D 94 B0
$9308: 3E 60 C6 24 10 EC AD 6D
$9310: 94 85 24 C6 24 A5 22 C5
$9318: 25 B0 02 C6 25 60 A4 24
$9320: A5 25 48 20 21 91 20 BD
$9328: 93 A0 00 68 69 00 C5 23
$9330: 90 F0 60 A5 22 85 25 20
$9338: 1F 91 A0 00 84 24 60 20
$9340: 33 93 A5 22 4C 22 93 A9
$9348: 00 85 24 E6 25 A5 25 C5
$9350: 23 B0 03 4C 21 91 C6 25
$9358: A5 22 48 20 21 91 A5 26
$9360: 85 FC A5 27 29 E3 85 FD
$9368: A5 28 85 2A A5 29 85 2B
$9370: 68 18 69 01 C5 23 B0 37
$9378: 48 20 21 91 A2 07 AD 6D
$9380: 94 4A A8 88 8D 55 C0 B1
$9388: 26 91 FC B1 28 91 2A 8D
$9390: 54 C0 B1 26 91 FC B1 28
$9398: 91 2A 88 10 E7 CA 30 BE
$93A0: 18 A5 27 69 04 85 27 A5
$93A8: FD 69 04 85 FD D0 CF 20
$93B0: B5 93 4C 1F 91 A0 00 2C
$93B8: A4 24 20 1F 91 A5 27 85
$93C0: FD 8C 6B 94 98 4A A8 8D
$93C8: 55 C0 90 03 8D 54 C0 A2
$93D0: 08 A9 00 4D 6F 94 91 26
$93D8: 18 A5 27 69 04 85 27 CA
$93E0: D0 EF A9 A0 24 32 30 02
$93E8: 49 80 91 28 8D 54 C0 A5
$93F0: FD 85 27 AC 6B 94 C8 CC
$93F8: 6D 94 90 C5 60 38 90 18
$9400: 6E 6E 94 60 A0 00 8C 6F
$9408: 94 88 84 32 60 A9 7F 8D
$9410: 6F 94 A9 3F 85 32 60 20
$9418: 2F FB 20 36 90 20 58 FC
$9420: 4C 06 90 F9 92 F5 92 F0
$9428: 92 F9 92 6D 90 F9 92 F9
$9430: 92 FA 92 09 93 F9 92 4A
$9438: 93 1D 93 3E 93 46 93 03
$9440: 94 0C 94 F9 92 FC 93 FE
$9448: 93 F9 92 F9 92 16 94 F9
$9450: 92 57 93 F9 92 32 93 B4
$9458: 93 F9 92 FF 92 B7 93 F9
$9460: 92 F9 92 00 00 00 00 00
$9468: 00 00 00 00 00 00 00 00
$9470: 00 00 00 00 00 AA AA AA
$9478: A0 A0 B6 B4 CB AD CB C1
$9480: D2 D4 C5 A0 C6 C5 C8 CC
$9488: D4 A0 A0 AA AA AA 87 87
$9490: 8D 8D 8C C4 EF F5 E2 EC
$9498: E5 AD C8 E9 F2 E5 F3 A0
$94A0: DA E5 E9 E3 E8 E5 EE E7
$94A8: E5 EE E5 F2 E1 F4 EF F2
$94B0: 8D D3 F4 E5 E6 E6 E5 EE
$94B8: A0 C8 EF EC FA E9 EE E7
$94C0: E5 F2 A0 B1 B9 B8 B5 AD
$94C8: B8 B6 8D 00 00 00 00 00
```

GERMAN.SET
Zeichensatz-Hex-Dump

```
$8D00: 00 00 00 00 00 00 00 00
$8D08: 08 08 08 08 08 00 08 00
$8D10: 14 14 14 00 00 00 00 00
$8D18: 14 14 3E 14 3E 14 14 00
$8D20: 08 3C 0A 1C 28 1E 08 00
$8D28: 06 26 10 08 04 32 30 00
$8D30: 04 0A 0A 04 2A 12 2C 00
$8D38: 08 08 08 00 00 00 00 00
$8D40: 08 04 02 02 02 04 08 00
$8D48: 08 10 20 20 20 10 08 00
$8D50: 08 2A 1C 08 1C 2A 08 00
$8D58: 00 08 08 3E 08 08 00 00
$8D60: 00 00 00 00 08 08 04 00
$8D68: 00 00 00 3E 00 00 00 00
$8D70: 00 00 00 00 00 00 08 00
$8D78: 00 20 10 08 04 02 00 00
$8D80: 1C 22 32 2A 26 22 1C 00
$8D88: 08 0C 08 08 08 08 1C 00
$8D90: 1C 22 20 18 04 02 3E 00
$8D98: 3E 20 10 18 20 22 1C 00
$8DA0: 10 18 14 12 3E 10 10 00
$8DA8: 3E 02 1E 20 20 22 1C 00
$8DB0: 38 04 02 1E 22 22 1C 00
$8DB8: 3E 20 10 08 04 04 04 00
$8DC0: 1C 22 22 1C 22 22 1C 00
$8DC8: 1C 22 22 3C 20 10 0E 00
$8DD0: 00 00 08 00 08 00 00 00
$8DD8: 00 00 08 00 08 08 04 00
$8DE0: 10 08 04 02 04 08 10 00
$8DE8: 00 00 3E 00 3E 00 00 00
$8DF0: 04 08 10 20 10 08 04 00
$8DF8: 1C 22 10 08 08 00 08 00
$8E00: 3C 02 1C 22 1C 20 1E 00
$8E08: 08 14 22 22 3E 22 22 00
$8E10: 1E 22 22 1E 22 22 1E 00
$8E18: 1C 22 02 02 02 22 1C 00
$8E20: 1E 22 22 22 22 22 1E 00
$8E28: 3E 02 02 1E 02 02 3E 00
$8E30: 3E 02 02 1E 02 02 02 00
$8E38: 3C 02 02 02 32 22 3C 00
$8E40: 22 22 22 3E 22 22 22 00
$8E48: 1C 08 08 08 08 08 1C 00
$8E50: 20 20 20 20 20 22 1C 00
$8E58: 22 12 0A 06 0A 12 22 00
$8E60: 02 02 02 02 02 02 3E 00
$8E68: 22 36 2A 2A 22 22 22 00
$8E70: 22 22 26 2A 32 22 22 00
$8E78: 1C 22 22 22 22 22 1C 00
$8E80: 1E 22 22 1E 02 02 02 00
$8E88: 1C 22 22 22 2A 12 2C 00
$8E90: 1E 22 22 1E 0A 12 22 00
$8E98: 1C 22 02 1C 20 22 1C 00
$8EA0: 3E 08 08 08 08 08 08 00
$8EA8: 22 22 22 22 22 22 1C 00
$8EB0: 22 22 22 22 22 14 08 00
$8EB8: 22 22 22 2A 2A 36 22 00
$8EC0: 22 22 14 08 14 22 22 00
$8EC8: 22 22 14 08 08 08 08 00
$8ED0: 3E 20 10 08 04 02 3E 00
$8ED8: 22 08 14 22 3E 22 22 00
$8EE0: 22 1C 22 22 22 22 1C 00
$8EE8: 22 00 22 22 22 22 1C 00
$8EF0: 00 00 08 14 22 00 00 00
$8EF8: 00 00 00 00 00 00 00 7F
$8F00: 04 08 10 00 00 00 00 00
$8F08: 00 00 1C 20 3C 22 3C 00
$8F10: 02 02 1E 22 22 22 1E 00
$8F18: 00 00 3C 02 02 02 3C 00
$8F20: 20 20 3C 22 22 22 3C 00
$8F28: 00 00 1C 22 3E 02 3C 00
$8F30: 18 24 04 1E 04 04 04 00
$8F38: 00 00 1C 22 22 3C 20 1C
$8F40: 02 02 1E 22 22 22 22 00
$8F48: 08 00 0C 08 08 08 08 1C 00
$8F50: 10 00 18 10 10 10 12 0C
$8F58: 02 02 22 12 0E 12 22 00
$8F60: 0C 08 08 08 08 08 1C 00
$8F68: 00 00 36 2A 2A 2A 22 00
$8F70: 00 00 1E 22 22 22 22 00
$8F78: 00 00 1C 22 22 22 1C 00
$8F80: 00 00 1E 22 22 1E 02 02
$8F88: 00 00 3C 22 22 3C 20 20
$8F90: 00 00 3A 06 02 02 02 00
$8F98: 00 00 3C 02 1C 20 1E 00
$8FA0: 04 04 1E 04 04 24 18 00
$8FA8: 00 00 22 22 22 32 2C 00
$8FB0: 00 00 22 22 22 14 08 00
$8FB8: 00 00 22 22 2A 2A 36 00
$8FC0: 00 00 22 14 08 14 22 00
$8FC8: 00 00 22 22 22 3C 20 1C
$8FD0: 00 00 3E 10 08 04 3E 00
$8FD8: 00 22 1C 20 3C 22 3C 00
$8FE0: 00 22 1C 22 22 22 1C 00
$8FE8: 00 22 00 22 22 32 2C 00
$8FF0: 1E 22 22 1E 22 22 1A 02
$8FF8: 00 2A 14 2A 14 2A 00 00
```

Apple ProDOS für Aufsteiger

Band 1, 2. Aufl., 203 S., DM 28,–
Band 2, 208 S., DM 30,–

von Ulrich Stiehl

Dr. Alfred Hüthig Verlag
Postf. 102869 · 6900 Heidelberg

Apple Assembler
Tips und Tricks

von Ulrich Stiehl
2. Aufl., 226 S., 3 Abb., kart., DM 34,–
ISBN 3-7785-1047-9

Dr. Alfred Hüthig Verlag
Postf. 102869 · 6900 Heidelberg

Utility-Programme für UCSD-Pascal

Disk-Editor, Library-Programm und Disassembler

von Dieter Geiß

Drei sehr umfangreiche Hilfsprogramme, die jedem Pascal-Programmierer wertvolle Dienste leisten, werden als Quelltexte auf die nächsten Peeker-Sammeldisketten verteilt werden. Es handelt sich dabei um

1. DISK (Disk-Editor),
2. LIBRARY (neue, verbesserte Version des Librarian) und
3. DISM (P-Code- und 6502/65C02-Disassembler).

Die Sammeldisk #22 (im DOS-3.3-Format!) zu dem vorliegenden Peeker-Heft enthält als erste Utility den Disk-Editor mit den Textfiles
DISK1.TEXT (30 Pascal-Blöcke)
DISK2.TEXT (36 Pascal-Blöcke)
DISK3.TEXT (36 Pascal-Blöcke)
Die zwei anderen Utilities erscheinen erst auf späteren Sammeldisketten. Deshalb beachten Sie bitte die entsprechenden Hinweise im jeweiligen Peeker-Heft. Das Library-Programm wird die Textfiles
LIBRARY1.TEXT (32 Pascal-Blöcke)
LIBRARY2.TEXT (30 Pascal-Blöcke)
umfassen, während sich die Textfiles
DISM1.TEXT (10 Pascal-Blöcke)
DISM2.TEXT (36 Pascal-Blöcke)
DISM3.TEXT (38 Pascal-Blöcke)
DISM4.TEXT (26 Pascal-Blöcke)
auf den P-Code-Disassembler beziehen werden. DISM wird außerdem das noch nicht erschienene Utility-Programm erfordern (siehe unten). Pascal-Anhänger mögen bitte dafür Verständnis haben, daß wir nicht alle Beiträge auf einmal bringen können, weil sich sonst die Applesoft- und Assembler-Anhänger benachteiligt fühlen. Die Beschreibungen von LIBRARY und DISM sind deshalb als Vorankündigungen zu verstehen.

Konvertierung und Compilierung

Wer die Programme benutzen möchte, muß sie zuerst mit GETDOS (auf Sammeldisk #15 enthalten) oder einem anderen DOS-Pascal-Transfer-Programm von der jeweiligen DOS-3.3-Sammeldisk (für den Disk-Editor ist dies Disk #22) auf seine eigene Pascal-Arbeitsdiskette konvertieren und dann compilieren lassen. Jedes der Programme ist in mehrere kleinere Textfiles (Angabe der Längen s.o.) aufgespalten, damit sie auch von denjenigen geändert werden können, die nicht über einen Editor verfügen, der Endlos-Texte verarbeiten kann. Zum Übersetzen müssen alle Files mit dem gleichen Präfix, also z.B.

DISK1.TEXT
DISK1.TEXT
DISK3.TEXT

auf die Diskette, auf der sich der Compiler befindet, oder die Boot-Diskette überspielt und dann mit z.B.

C(ompile DISK1

compiliert werden. Als Codefile kann man <Return> für SYSTEM.WRK.CODE oder DISK, was dann z.B. für DISK.CODE steht, angeben. Wenn die Speicherkapazität bei 140K-Disketten nicht ausreicht, so gebe man für den Codefile #5:DISK an (also Quelltexte auf #4 und Objektcode auf #5).
Tritt während des Übersetzens ein Stack-Overflow auf (nur bei Pascal 1.1 oder 1.2 64K), so muß dem zu übersetzenden File die Compileroption (*$S+*) vorangestellt werden.
Da die Quelltexte verfügbar sind, kann jeder die Programme zu seinem eigenen Zweck abändern.

1. Bedienung von DISK

Nach dem Starten erscheint eine Kommandozeile, die 8 von 23 Kommandos anzeigt. Weitere Kommandos sind wie beim Filer durch Eingabe eines Fragezeichens (?) zu sehen. Ein Kommando kann auch gegeben werden, wenn es nicht in der Kommandozeile zu sehen ist.

Es folgen die zwei Statuszeilen, die angeben, welche Unit, welcher Block und welcher Modus für die beiden „Fenster" gelten. In der Kommandozeile wird am rechten Rand das gerade aktive Fenster angezeigt. Es kann 1, 2 oder 1 + 2 sein. Beide Fenster können *gleichzeitig* nur mit einer angeschlossenen UltraTerm-Karte gesehen werden (80x48-Modus).

Unter den Statuszeilen erscheint der Rahmen, in welchem die beiden Sektoren eines eingelesenen Blocks angezeigt werden. Die Adressen sind in hexadezimaler Schreibweise angegeben.

Durch Eingabe des Anfangsbuchstabens wird ein Kommando ausgeführt (Ausnahme: eXchange wird durch X ausgeführt). Jedes Kommando kann durch Eingabe von <Esc> vorzeitig abgebrochen werden.

Unit: Die Unitnummer für das aktive Fenster wird erfragt. Es sind nur blockstrukturierte Geräte zugelassen.

Block: Die Blocknummer für das aktive Fenster wird erfragt.

Read: Ein Block wird eingelesen und im aktiven Fenster angezeigt. Sind beide Fenster aktiv, wird zuerst Fenster 1, dann Fenster 2 angezeigt, wobei Fenster 1 wieder überschrieben wird. Nur bei angeschlossener UltraTerm können beide Fenster gleichzeitig gesehen werden.

Write: Der im Fenster angezeigte Block wird auf die aktuelle Unitnummer auf den aktuellen Block geschrieben.
Edit: Ein Block kann editiert werden. Es darf nur ein Fenster aktiv sein. Je nach Modus (ASCII, Hex, Mixed) können ASCII-Zeichen, Hex-Zahlen oder beide editiert werden. Im ASCII-Modus werden alle Zeichen als ASCII-Zeichen angenommen, im Hex-Modus nur gültige Hex-Ziffern, im Mixed-Modus beide. Dort wird außerdem, wenn man eine Zahl zwischen $20 und $7F eingegeben hat, diese sofort in das entsprechende ASCII-Zeichen umgewandelt und angezeigt. Mit den Pfeiltasten kann man sich bewegen, wie man es im Editor gewohnt ist.
<Ctrl-I> springt zur nächsten Tabulatorposition (alle 4 Positionen),
<CR> an den Anfang der nächsten Zeile.
Mit <Esc> erreicht man ein vorzeitiges Verlassen, wenn man die Daten doch nicht ändern will. Die alten Daten werden dann wieder angezeigt, falls etwas verändert wurde. Verläßt man den Editor mit
<Ctrl-C>, so werden die veränderten Daten übernommen. Durch ein nachfolgendes Write können sie dann auf Diskette abgespeichert werden.
ASCII: Sind schon gültige Daten im Puffer (mindestens einmal muß etwas eingelesen worden sein), werden sie im ASCII-Modus angezeigt. Alle nicht druckbaren Zeichen werden als „-" ausgegeben.

Hex: Die Daten werden im Hex-Modus angezeigt.
Mixed: Die Daten werden gemischt angezeigt, d.h. alle druckbaren Zeichen als ASCII-Zeichen, der Rest als Hex-Zahlen.

Search: Es kann nach einem Muster gesucht werden. Es wird zuerst die Blockuntergrenze (Lower bound), dann die Blockobergrenze (Upper bound) und schließlich das Muster erfragt. Im ASCII- und Mixed-Modus ist es eine Zeichenkette, die durch <Return> abgeschlossen wird. Im Hex-Modus kann ein Hex-Muster eingegeben werden (z.B. 10 3D A9 01). Die dazwischenliegenden Leerzeichen müssen nicht eingegeben werden. Man kann also sofort z.B. 103DA901 eingeben. In der Anzeige werden die Leerzeichen mit dargestellt. Muß man eine 01 eingeben, so kann man entweder direkt 01 oder auch <Space>1 schreiben. Danach wird der angegebene Bereich untersucht. Jeder Punkt, der nunmehr ausgegeben wird, bezeichnet einen Block. Bei positivem Ergebnis werden Blocknummer und Bytenummer angegeben. Danach kann man sich entscheiden, ob weitergesucht werden soll.

eXchange: Ein Muster wird gegen ein anderes ausgetauscht. Das Zielmuster muß kürzer oder genauso lang sein. Es gilt das gleiche wie für Search, nur daß nach dem Suchmuster noch ein Zielmuster einzugeben ist. Außerdem gibt es die Möglichkeit, alle oder nur einige Muster auszutauschen.
Increment: Die Blocknummer wird erhöht, danach wird ein Read oder Write durchgeführt, je nachdem, welche Operation zuletzt durchgeführt worden ist.
Decrement: Wie Increment, wobei die Blocknummer zuerst erniedrigt wird.
1, 2, 3: Legt das aktuelle Fenster fest. Es wird in der Kommandozeile rechts oben angezeigt. 3 bedeutet beide Fenster, die aber nur mit angeschlossener UltraTerm gleichzeitig dargestellt werden können.
Copy: Ein blockweises Kopieren kann vorgenommen werden. Es wird von <1> nach <2> kopiert. Die Anzahl der Blöcke sowie eine Option, ob die Blöcke gleichzeitig angezeigt werden sollen, können bestimmt werden.
Verify: Eine bestimmte Anzahl von Blökken von <1> kann mit <2> verglichen werden. Alle Bytes, die gleich sind, werden als „-" angezeigt, die verschiedenen Bytes mit „x".
Lines: Der UltraTerm-Modus kann gewählt werden. Die Zahl, die angegeben werden kann, entspricht dem Video-Modus, wie er auf Seite 5.3 des UltraTerm-Handbuchs angegeben ist. Dabei spielt Modus 6 eine besondere Rolle. Mit ihm hat man 48 Zeilen zur Verfügung, so daß beide Fenster gleichzeitig dargestellt werden können.
Get file: Ein Directory kann von einem blockstrukturierten Gerät eingelesen werden. Es wird die erste Datei angezeigt, und mit den Pfeiltasten können dann die anderen Dateien angezeigt werden. Ein „y" holt die Blockadresse der Datei. Ein anschließendes Read liest dann den ersten Block ein.
Trick: Wird „T" betätigt, so wird der Trick-Modus eingeschaltet. In diesem Modus können die EXPORT-Texte von Modula-2-Modulen gelesen werden, die vom Modula-2-Compiler (Version 0.3a) verschlüsselt werden. Ein nochmaliges „T" schaltet diesen Modus wieder aus.
Fill: Ein Bereich eines Fensters kann mit einem festen Wert (z.B. 0) gefüllt werden. Es treten die Fragen nach dem „Start index", dem „End index" und dem „Hex value" auf.
Quit: Verläßt den Disk-Editor.

2. Bedienung von LIBRARY

Sie entspricht im wesentlichen der Bedienung des Originalprogramms. Auch hier können wie in DISK durch Eingabe eines Fragezeichens weitere Kommandos sichtbar gemacht werden. Zuerst erscheint die Frage nach dem
„Output code file".
An dieser Stelle kann neben einem Namen auch „*", <Esc> <Return> oder <Return> eingegeben werden. „*" steht für *SYSTEM.LIBRARY, ansonsten steht ein Stern für SYSTEM.LIBRARY, z.B. #5:* = #5:SYTSEM.LIBRARY. Mit <Esc> <Return> kann das Programm an dieser Stelle vorzeitig verlassen werden. Gibt man <Return> ein, so wird in einen speziellen Teil verzweigt, der weiter unten erklärt wird.
Auf die Frage
„Link code file"
kann ebenfalls „*", aber auch „$" oder <Return> eingegeben werden. Der Stern hat dieselbe Bedeutung wie oben, „$" bezieht sich auf den gleichen Namen wie der „Output code file". Wird der Codefile nicht gefunden, so wird ein Suffix „.CODE" angehängt und der Versuch wiederholt.

Jetzt werden links die Slots des „Link code files" und rechts die des „Output code files" angezeigt. Folgende Kommandos sind nun möglich:
<Zahl>: Bei Eingabe einer Zahl zwischen 0 und 15 mit abschließendem <Return> oder <Space> kann dieser Slot kopiert werden. Es erscheint die Frage
„Slot to link into?",
die ebenfalls eine Antwort zwischen 0 und 15 erwartet. Ist der Slot besetzt, wird ein Warnung ausgegeben. Mit <Esc> kann jeweils vorzeitig abgebrochen werden.
„=": Alle Slots werden vom „Link code file" in den „Output code file" kopiert.
Select: Ein selektives Kopieren kann durchgeführt werden. Alle gefüllten Slots werden durchgefragt. Ein „y" kopiert, ein <Esc> verläßt vorzeitig diesen Modus. Alle anderen Zeichen kopieren nicht.
New file: Ein neuer „Link code file" kann angegeben werden.
Quit: Das Programm wird ordnungsgemäß verlassen. Eine Notiz kann noch in die Library geschrieben werden.
Abort: Das Programm wird abgebrochen. Alle Files befinden sich wieder in dem Zustand, der vor dem Starten des Programms herrschte.
Link: Ist Link eingeschaltet, so können Intrinsic Units, die kein Datensegment besitzen, zum Codefile hinzugelinkt werden. Dabei entsteht ein sofort ausführbares Programm, welches keine Library benötigt. Ein Beispiel für ein solches Programm ist der SYSTEM.ASSMBLER. In seinem Codefile befindet sich die Unit PASCALIO, die normalerweise in der Library sitzt. Damit der Assembler auch ohne SYSTEM.LIBRARY abläuft, wurde die Unit zum Codefile hinzugelinkt. Das kann man nicht mit dem Linker durchführen, da es sich um eine Intrinsic Unit handelt. Damit das Pas-

cal-System danach aber ordnungsgemäß den fertigen File ausführen kann, muß der Status des Segments noch mit Hilfe des speziellen Teils umgewandelt werden (siehe LInt→Lnkd). Der Anfangszustand ist <aus>.
Inter: Mit Hilfe dieses Flags kann bestimmt werden, ob der Interface-Teil mit dem Codefile kopiert wird. Der Anfangszustand ist <ein>.
Code: Ist „Code" eingeschaltet, wird der Code der Eingabe-Slots kopiert, sonst nicht. Der Anfangszustand ist <ein>.
View: Zeigt den Status der Flags.
Fill slots: Füllt die leeren Slots des „Output code files" der Reihe nach mit den gefüllten Slots des „Link code file".
Warning: Das Flag gibt an, ob Warnungen ausgegeben werden sollen.
Zero: Löscht das „Set of intrinsics", das von einem Programm benutzt wird.
Help: Bietet alle Kommandos an, wobei nach Wählen eines Kommandos eine Hilfe-Zeile am untersten Bildschirmrand erscheint.

Der oben erwähnte spezielle Teil ist zum Untersuchen von Codefiles bestimmt. Angezeigt werden:
– Segmentnummer,
– Segmentname,
– Adresse des Segments,
– Länge des Segments,
– Art des Segments,
– Textadresse für Units,
– Maschinentyp und
– Version.
Eine neue Kommandozeile erscheint:
Clear: Löscht die Anzeige und fragt nach einem neuen File.
Pascal: Nimmt als Eingabefile *SYSTEM.PASCAL.
System.wrk.code: Nimmt als Eingabefile *SYSTEM.WRK.CODE.
New file: Eine neue Eingabedatei kann spezifiziert werden.
LInt→Lnkd: Alle Segmente, die vom Typ „Linked intrinsic" sind, werden in „Linked" umgewandelt. Ein solcher Codefile kann sofort ausgeführt werden (siehe auch SYSTEM.ASSMBLER).
Quit: Verläßt den speziellen Teil und damit das gesamte Programm.

3. Bedienung von DISM

Zum Untersuchen von Codefiles ist es manchmal angebracht, sie zu disassemblieren. DISM gibt den P-Code als mnemotechnische Kürzel aus. Standardprozeduren und Level-0-Prozeduren werden zusätzlich angezeigt. Außerdem werden alle Standardprozeduren von Modula-2 (Version 0.3a) ebenfalls mit ausgegeben. Des weiteren wird 6502- und 65C02-Code disassembliert.

Wer das Programm übersetzen läßt, muß dafür Sorge tragen, daß sich die beiden benutzten Units „Applestuff" und „Utility" in der Library befinden.

Redaktioneller Hinweis: Die Unit „Utility" wird im Teil 8 der Peeker-Serie „Tips und Tricks in Pascal" vorgestellt werden und erscheint in Verbindung mit der Sammeldisk, die DISK enthält. Die Teil 7 und 8 der Serie sind bereits redaktionell bearbeitet, doch aus Platzgründen noch nicht erschienen.

Nach dem Starten erscheint eine Kommandozeile mit 8 Befehlen. Weitere 5 Befehle werden durch Eingabe eines Fragezeichens (?) sichtbar gemacht.
Input: Der Eingabefile wird festgelegt. Bei Eingabe von <Return> wird automatisch der Workfile genommen. Mit <Esc> <Return> kann man diesen Punkt verlassen. Wird ein Fragezeichen, gefolgt von einem <Return>, eingegeben, so kann man das Inhaltsverzeichnis einer „Unit" abfragen (siehe Prozedur ListDirectory in der Unit „Utility").
Output: Der Outputfile wird festgelegt. <Return> steht für CONSOLE: (= Bildschirm).
Memory: Der Speicher kann disassembliert werden.
Pcode: Der P-Code wird gelistet. Die Adresse kann vorher über „Address" eingegeben werden. Nach dem Laden eines neuen Programms wird die Anfangsadresse automatisch auf $0200 gestellt, was normalerweise der Anfangsadresse des Codes entspricht. Nach einer Prozedur wird die Prozedurnummer, Verschachtelungstiefe, Enter IC, Exit IC und die Sprungtabelle (Jump Table) ausgegeben. Nach dem Hauptprogramm werden auch die Zeiger auf die einzelnen Prozeduren ausgegeben. Dazu muß vorher die Anzahl der Prozeduren genannt werden, die das Programm deshalb nicht weiß, weil der Code vielleicht nicht von Anfang an gelistet wurde. Es wird aber eine Zahl vorgeschlagen, die der Anzahl der Prozeduren entsprechen könnte und die höchste bis dahin gefundene Prozedurnummer ist. Bei Eingabe von <Esc> wird der List-Vorgang abgebrochen. Wird dann noch einmal „Pcode" aufgerufen, so wird an der Stelle weitergemacht, wo abgebrochen wurde.
6502: 6502- und 65C02-Code wird gelistet. Es gilt das gleiche wie für Pcode, nur daß keine P-Code-spezifischen Dinge ausgegeben werden.
Dump: Es wird ein Hex- und ASCII-Dump durchgeführt. Ansonsten gilt das gleiche wie für 6502.
Address: Die Anfangs- und Endadresse kann bestimmt werden.
Quit: Verläßt das Programm.

Computed Address: Will man gezielt eine Prozedur eines Segments disassemblieren lassen, so wählt man „Computed Address". Es erscheinen alle Segmente des Eingabefiles. Mit dem entsprechenden Buchstaben wählt man ein Segment aus. Dann muß man noch die Prozedurnummer bestimmen. Die Anfangsadresse wird auf diese Prozedur eingestellt. Ein nachfolgender „P-Code"-Befehl listet die Prozedur und alle folgenden auf.
Block: Damit kann eine zum Anfang des Files relative Blocknummer angegeben werden. Block 0 ist der erste Block usw.
Syntax: Die Ausgabe beim 6502 kann auf UCSD- oder MOS-Schreibweise eingestellt werden. Beispiel:
UCSD: JMP §0FF00
MOS: JMP ($FF00)
Upper case: Es erscheint die Frage „Has lower case?", die mit „n" zu beantworten ist, wenn man nur über einen Apple II Plus ohne 80-Zeichenkarte verfügt. Dann werden Kleinbuchstaben im P-Code in Hex-Darstellung gegeben. Beim Starten des Programms wird angenommen, daß ein Bildschirm vorhanden ist, der Kleinbuchstaben erzeugen kann.
Xspecial: Ist der Special-Modus eingeschaltet (Default), so werden bei „Pcode" Jump Table, Enter IC, Exit IC, Data Size, Parameter Size, Prozedurnummer und Verschachtelungstiefe ausgegeben. Durch Drücken von „X" kann man diesen Modus unterbinden, durch nochmaliges Drücken wieder einschalten.

APPLE & CP/M-80 & MS-DOS SOFTWARE & HARDWARE

z. B. für APPLE II und Kompatible
Wir liefern die RAM-Karte (AE) für den Apple IIe mit max. 3 MB (Appleworks mit mehr als 2 MB)! 64-K-Ausf. ..... DM 650,–
SpeeDemon 3.56 MHz Coproz. für II+/e (McT) ..... DM 700,–
Anpassung für Appleworks 1.2 auf dem II+/e.
Original oder mit externer Tastatur. Anpassung in deutsch für SATURN 128 K und IBS AP33 1 MB! ..... DM 170,–
UPC-Programmer-Card 2716-128 komfortabel ..... DM 580,–
72 I/O Port Card programmierbar DOS+CP/M ..... DM 350,–
AD 16 Ch. 12 Bit, schnell! (Applied Eng.) ..... DM 1150,–
PKASO/U-Printer-Karte (IS) ..... DM 550,–
CP/M-Plus-Card, 6 MHz, 64 K, CP/M 3.0 (ALS) ..... DM 1150,–
Timemaster II H.O., die Uhrenkarte! (AE) ..... DM 540,–
ELF kompl. Statistik-Software (TWG) ..... DM 800,–
Prime-Plotter-Grafik-Software (Primesoft) ..... DM 900,–
Z-RAM 512 K für APPLE IIc (AE) ..... DM 1250,–

z. B. für IBM und Kompatible
APPLE Turnover (Vertex) Lesen/Schreiben von Apple Disks im IBM PC & Komp. ..... DM 1200,–
XENO-COPY plus (Vertex) Lesen/Schreiben div. CP/M & MS-DOS Formate im IBM ..... DM 600,–
ELF PC kompl. Statistik Software (TWG) ..... DM 800,–
PROM Blaster 28-Pin (Apparat Inc.) ..... DM 620,–

z. B. für alle Systeme
Printerchanger 3 parall. Drucker auf 1 Micro inkl. Kabel/Netzteil (Keyzone) ..... DM 570,–
Printersharer 3 Micros auf 1 parall. Drucker inkl. Kabel/Netzteil (Keyzone) ..... DM 460,–
Shufflebuffer 64 K (IS) ..... DM 1250,–
Wir sind Import-Spezialisten und bieten Ihnen eine große Auswahl an Software und Hardware bedeutender Hersteller aus den USA und England.
Informationen gegen DM 3,– in Briefmarken.

WEISS COMPUTER Dipl.-Psych. Karl-Heinz Weiß
Am Wiesenhof 17, 2940 Wilhelmshaven, Tel. 0 44 21/8 31 79

Zubehör für APPLE II, IIe

II+ komp. Mainboard 48K o. Firmware	388,-
Disk-Controller für Shugart-BUS - 35/40 Track Laufwerke - 80 Track Laufwerke - gemischter Betrieb möglich - inkl. Handbuch und Software	149,-
80 Zeichenkarte m. SSW - 4 Zeichensätze (2 * 2732) - Hell-/Dunkel-Steuerung möglich	178,-
ACCELERATOR 3,6MHz 6502C für II+	358,-
Super-Serielle-Karte vollduplex	109,-
WILD-Karte (knackt Programme)	68,-
AD/DA-Wandler 8 Bit/16 Kanäle	259,-
TURBO-PASCAL 2.0 u. TOOLBOX je	98,-

Alle Karten aus eigener Fertigung und mit hochwertigen IC-Fassungen und 1.Wahl Bauelementen bestückt. Keine Taiwanware!!!

mp//c-703 Tastatur

488,-

Ultraflache ASCII-codierte Tastatur für alle APPLE II PLUS und kompatible Computer. Die Tasten sind mit Goldkontakten und 100% abriebfesten Kappen versehen. 19 Funktions-, Cursor-, Tab- und Del-Tasten können in drei Ebenen mit je 6 ASCII-Zeichen belegt werden. Sonderbelegungen und Anpassung an andere Rechner ist durch unseren Kundendienst möglich! Für ausführliches Info bitte Prospekt anfordern. Bitte DM 1,40 für Porto beifügen.

mp//c-Datentechnik Postfach 4248 5014 Kerpen 4

Besuchen Sie unsere Geschäftsräume in 5014 Kerpen-Brüggen, Heerstr. 392 von Di.-Fr. 9.00-12.00 Uhr u. 14.00-17.00 Uhr
APPLE II+, IIe sind eingetragene Warenzeichen der Fa. APPLE Computer

Ihr direkter Draht zur Anzeigenabteilung
Tel.: 06221 / 489206

Erster aaa „little big" Ableger

für Apple II/+/e, Basis 108 und compatible

Sicher. – Wir haben uns bei der Entwicklung u. Fertigstellung unserer **Mega Ramcard II**® etwas Zeit gelassen, damit Sie jetzt auch Zeit gewinnen. Und das hat sich gelohnt!
Die Tests, aber auch die mitgelieferten Ramdisktreiber-Programme für folgende Betriebssysteme, beweisen es:

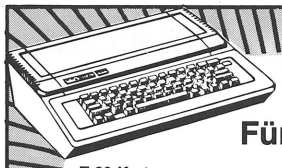
- Apple DOS 3.3
- Diversi DOS 2-C/4-C
- Apple ProDOS 1.0.1, 1.1.1
- Apple Pascal 1.1
  Apple Pascal 1.2/64 K, 1.2/128 K
  Apple Pascal 1.3/64 K, 1.3/128 K
- CP/M 2.2 für
  Microsoft Softcard, CP/M 2.20, 2.23
  Microsoft premium card, CP/M 2.26
  Microsoft Softcard II, CP/M 2.28

Sogar nach einem „Systemabsturz" oder Betriebssystemwechsel bleiben alle Dateien erhalten! Peripherieslot und Speicherkapazität werden automatisch erkannt, sodaß das lästige Eingeben entfällt.
Mega Ramcard II® ist mit den erphi-AFDC2/AFDC3-Controller vollständig kompatibel, u.v.m.
Demo-Diskette = 4,50 (in Briefmarken) anfordern.
Händleranfragen schriftlich erwünscht.

Mega Ramcard II, 1 MByte bestückt + Software (s.o.) + deutsches Handbuch, II/+/e	**878,–**
Mega Ramcard II, 256 K-RAM best. (bis 1 MB aufr.) + Software (s.o.) + dt. Hb., II/+/e	**498,–**
Disk II-Siemens-Laufwerk im Geh.+ Kabel f. orig. + comp. Contr. geeignet, II/+/e	**298,–**
Disk II-Siemens-Laufw.+ Contr.+ Kabel, II/+/e	**359,–**
Erphi-AFDC2-Controller + Autopatch Software + deutsches Handbuch, II/+/e	**198,–**
Disk II-Philips Laufwerk o. Geh., 2 x 80 Track-640 KB f. AFDC2 o. 3 modifiziert, II/+/e	**298,–**
Erphi-DuoDisk 1,2 MByte im Gehäuse + Erphi-AFDC3-Controller, II/+/e	**898,–**

REPARATUREN an Apple + Compatiblen Geräten + Zubehör führt unser Spezialteam garantiert zuverlässig + besonders kostengünstig aus. **Sprechen Sie mit uns. Kostenvoranschlag auf Wunsch!**

aaa electronic gmbh
Telex 0772642 aaa-d
Habsburgerstraße 134
7800 FREIBURG, Tel. (0761) 276864
Bauelemente – Bausätze – µP's
Meßgeräte – Zubehör – Fachliteratur
Fachgeschäft für Elektronik + Mikrocomputer



Für Apple II, IIe

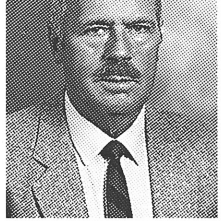
Z-80-Karte	69,–	**80-Zeichen-Karte** mit Softswitch, nur für II Plus kompat.	149,–
Disk-Interface	69,–		
Centronics-Interf. m. Kabel	69,–	**Speech-Karte**	55,–
16-K-Ram-Karte	79,–	**Clock-Karte**	99,–
128-K-RAM-Karte	199,–	**Komp 2E** Apple 2E kompatible. Rechner 64K im 2E-Design, ohne Firmware.	699,–
256-KB-RAM-Karte	299,–		
Motherboard 64K II Plus kompat. 6502, Z80, 64K ohne Firmware	292,–	**80Z + 64K-Karte** für 2E kompatible.	99,–
		Motherboard 2E 2E kompatible ohne Firmware.	299,–

Händleranfragen erwünscht!
Apple-Info 1,– DM (Porto)

Klaus Jeschke
Hard-, Software
Viertstr. 3-13
6233 Kelkheim
☎ (0 61 98) 90 69

Sicherheit auf allen Wegen

Dazu: Willi J. Alef, Kaufmann, Lippstadt



Meine Meinung ist:
Wenn ich bei Freunden mal feuchtfröhlich feiern will, erkundige ich mich zuvor nach einem sicheren Heimweg mit Bus oder Bahn. Oder ich fahre mit dem Taxi. Dann kann ich ruhig ein paar Bierchen trinken und komme trotzdem sicher nach Hause. Willi J. Alef

BG Hallo Partner danke schön

Ihre Berufsgenossenschaft und der Deutsche Verkehrssicherheitsrat

EcoSoft
Economy Software

Kaiserstraße 21, 7890 Waldshut, Telefon 07751-7920

Software (fast) gratis

über 20.000 Frei-Programme für IBM-PC/Kompatible, Apple II, Macintosh, Commodore C64/C128, Amiga, Atari ST.

Textverarbeitung, Datenbanken, Rechenblätter, Finanzprogramme, Buchhaltungen, Börse, Programmiersprachen und -Hilfen, Lernprogramme, CAD, Graphic, Spiele, Heimprogramme, Musik, usw.

● **Verzeichnis gratis**

Bitte Computer-Modell angeben.

aaa electronic gmbh

Neue Preise # 10

IBM-PC/XT+AT-COMPATIBLE COMPUTER + ZUBEHÖR (X=XT, A=AT, X/A=XT+AT)	LEER/BESTÜCK
Mainboard XT-256 f. 16 bit CPU, OK (bis 256K-on board, oder 640 K mit Addr. auf Karte aufr.) + Boot-ROM + 8 Slots gepruft	79,--/279,--
Mainboard XT-256 mit 256 K-RAM, sonst. w.o., gepr.	329,--
Mainboard XT-640 f. 16 bit CPU, OK (bis 640 K-on board) gepr	89,--/288,--
Mainboard XT-640 w.o. jedoch 256 K-RAM best. + gepr.	365,--
Mainboard XT-640 w.o. jedoch 640 K – vollbest. und gepruft	438,--
Mainboard TURBO XT-640 – 4/7/8 MHz, OK, best. + gepr.	109,--/339,--
Mainboard AT-1MB, CPU 80286 16/24 bit (mit Sockel f. 80287) 512 K-RAM best. (bis 1MB aufrüstbar) + Boot-ROM + Uhr mit Akku + 8 Slots, gepr.	**1545,--**
64KB-RAM-Chip-Aufrüstsatz (9 Stück) einzeln gepruft! **X/A**	37,--
256 KB-RAM-Chip-Aufrüstung (9 Stück) einzeln gepruft! **X**=89,--/**A**=99,--	
Disk Controller gepr. (f. 4x360 KB-Drives)(Extra Kabel=39,-)**X**=34,--/94,--	
Harddisk + Floppy-Controller (f. HD + 2DD) gepr., **A**	697,--
Disk-Controller gepr. (2x1,2 MB-Drives), **A**	198,--
Monochrome Karte gepruft, **X/A**	49,--/169,--
Monochrome Graph.+par.Print. Karte (Hercules comp.), **X/A**	55,--/188,--
Color RGB + Video Graphikkarte+Lightpen-Ausg., gepr., **X/A**	55,--/153,--
Parallel Printer-Karte gepr. o. Kabel (Extra Kabel = 49,–), **X/A**	29,--/ 56,--
RS 232 Interface (2xRS 232, 1 davon bestückt), **X/A**	29,--/ 83,--
RS 232+par. Printer Karte gepr., **X/A**	147,--
512K-RAM-Karte gepr. (OK)(mit 64 KB-RAM bestückbar), **X**	55,--/117,--
3 MB-RAM-Erweiterungsk. gepr. (OK), (mit 256 K-RAM bestückt), **A**	399,--
2,5 MB-Multifunktionskarte, OK, + RS 232,+ par. Printer, gepr., **A**	445,--
384K-Multifunktionskarte, RS 232, par. Printer, Intert. Game/Joystick Port, Real Time Clock gepuft, OK + RAM-Disk/Spooler/Uhr Softw., **X**	55,--/224,--
Disk Multi I/O Karte (f. 2x360 K-Drive, 2xRS 232, 1xpar. Printer Int., Game/Joystick Port, Real Time Clock/Kalender gepuft), **X**	59,--/225,--
Disk I/O w.o. sowie Color-Graphik/RGB + Composite + Monochrome TTL-Karte gepr., **X**	489,--
AGA-Karte + Software + Hb. dt., Commodore original, **X/A**	639,--
EGA-comp. Enhanced Graphic Adapter Karte, mit 256 K-RAM-Bildschirmspeicher, Monochrome-Color Graphik (640 x 350 Punkte – 16 Farben) + par. Printer Int. voll best. + gepr., **X/A**	69,--/685,--
Prototype/Lochraster Leerkarte, durchkontaktiert, lang	64,--
Prof.-Funktionstastatur ASCII + 15er Block ab **195,---**	/dt. ab 225,--
135W/150W-Netzteil mit eingebautem Ventilator, kurzschlußf.	**195,**--/215,--
200W-Netzteil mit eing. Ventilator, kurzschlußfest, **A**	375,--
360KB-Diskdrive Slimline (Track Access Time < 2 mS), R/W gepr., **X/A**	279,--
720KB-Diskdrive Slimline, 2x80 Tr., R/W gepr., **X/A**	339,--
1,2MB-Diskdrive Slimline, 2x80 Tr., R/W gepr., **X/A**	379,--
20MB-Harddisk, X/A	1248,--
20 MB-NEC-Festplatte + DTC-Controller mit Kabel + Man., gepr., **X**	1.438,--
Joystick f. IBM + Compatible, X/A	**79,**--
Maus incl. Software f. IBM+Compati. (MOUSE Systems comp.), **X/A**	197,--
Met.-Gehäuse aufkl. (Pl.f.4.-Einb.-Drives), solide Ausf., **X**=198,--/**A**=298,--	
PC/XT + AT – KOMPLETT-SYSTEME AUFGEBAUT UND GEPRÜFT	
XT-640, 16 Bit-Rechner (incl. Boot-Eprom) 256K-Ram im IBM-look-like Met.-Gehäuse+Profi-Fkt. Tast.+135 W-Netzt. gepr.	**999,--**
XT-640, w.o. + Contr. + 360KBDrive + Color RGB + Vid. Graphik-Karte	1.169,--
XT-640, w.o. jedoch statt Color mit Monochrome Karte	1.179,--
XT-640, w.o. jedoch mit Monochrome-Graphik-Printer-Karte	1.189,--
XT-640, 256 K-RAM + 360 KB Drive + Disk Multi I/O + Color-Gr.-Karte	1.379,--
XT-640, 256 K-RAM + 360 KB Drive + Color RGB + Video-Graphik-Karte + 20 MB-NEC-Festplatte + DTC-Controller	2.599,--
AT-1MB, CPU 80286 512 K-best., im IBM-look-like Met.-Gehäuse + Profi-Fkt. Tast + 200W-Netzteil + Color Graph.-Karte + 1,2MB-Drive, gepr.	**2879,--**
AT-1MB, w.o. + 20MB-Harddisk gepr.	4385,--
Aufpreis für Turbo XT-640 -4/7/8 MHZ umschaltbares Board, **X**	50,--
Aufpreis für XT-640 M-board mit 640 K vollbestückt + gepr., **X**	**159,--**
Aufpreis f. EGA anstatt Color Grafik Karte, **X/A**	549,--
PC 10-512 K + 1 Laufwerk + 20 MB-Festplatte	**5549,--**

APPLE-BUS COMPUTERPLATINEN + PERIPHERIE	LEER/BESTÜCKT
M-board II 48K, gesockelt, vollbest. + gepr.	57,--/337,--
M-board II 64K gesockelt, vollbest.+gepr.	88,--/375,--
M-board II 64K + Z80 CPU, gesockelt, vollbest. + gepr.	95,--/378,--
M-board IIe 64K vollbestückt + gepr.	98,--/389,--
Superstarkes Netzteil – kurzschlußfest 5A=**115,---** b.z.w. 7A=	145,--
Entwicklungslochraster-Leerplatine durchkontaktiert	27,--
16K-Speicher-Karte geprüft	18,95/ 55,--
APPLE orig. 16K-Language Karte gepr.	98,--
Z-80 CPU-Karte geprüft (CP/M Vol. I.+II Manual dafür = 69,--)	18,95/ 64,--
Controller DOS 3.3-Karte geprüft	18,95/ 63,--
Auto-Controller DOS 3.2/3.3-Karte gepr.	24,--/ 97,--
Erphi AFDC 2 Controller + Autopatch Softw. + Handbuch, gepr.	198,--
Parallel-Drucker-Karte geprüft (Extra Kabel = 45,--)	18,95/ 74,--
Par.-Drucker Karte-Grappler comp. gepr. (Extra Kabel=45,--)	27,--/169,--
8bit par. Int. + 64K-Buffer + Kabel gepr. (Grappler comp.)	39,--/299,--
Seriell-Interface-Karte V24 geprüft	18,95/ 88,--
Super-Seriell-Interface Karte gepr.	34,--/169,--
APPLE-Original Super-Seriell-Interface gepr.	295,--
80-Zeichen/24-Zeilen-Karte geprüft	18,95/119,--
80Z/24Z-Karte + Softwitch-Schalter, gestochen scharf, gepr.	24,--/127,--
80Z/24Z-Softwitch Karte gepr. für Ile	15,--/ 39,--
80Z/24Z + 64K-Ram + Softw.-Karte best. + gepr. f. Ile	18,95/ 87,--
IEEE-488 Int. Karte gepr. (Extra Kabel=45,--)	29,--/297,--
PAL-Modulator/Color-Karte + UHF Modulator, geprüft	18,95/ 99,--
RGB-Color Karte gepr., (nur f. II+)	18,95/129,--
128K-Speicher-Leerkarte + Software + Manual	99,--
128K-Speicher-Karte + Software + Manual, gepruft	29,--/255,--
Mega Ramcard II, 256 KB best. + Software + dt. Hb. (bis 1 MB aufr.)	498,--
Mega Ramcard II, 1 MB best. + Software + dt. Hb., gepr.	878,--
EPROM-Burner (2716/32/64) geprüft	24,--/109,--
6522-VIA Karte geprüft	24,--/135,--
Clock Karte + Software + Manual, geprüft	24,--/ 94,--
Speech Karte + Software + Manual, geprüft	18,95/ 53,--
Wild Karte + Software + Manual (nicht f. IIe), geprüft	18,95/ 59,--
AD/DA-8Bit Karte + Software + Manual, gepr.	29,--/375,--
UHF/TV-Modulator universell	27,50
Lüfter anclipsbar (220 V)	79,--

APOLLO//(48K) + UHF-Mod. + Gr/KI vollbest. + gepr.	617,--
APOLLO//(48K) + Disk II F + Contr. + 12"-Monitor	1.285,--
APOLLO//A (48K) + Gr/KI + 15er-Tastatur vollbest. + gepr.	677,--
APOLLO// ASKF (48K) vollbest. + gepr sep.Keyboard + Gr/KI + 15er-Block mit Funkt.-Tasten im „IBM-look like" Kunststoff-Gehäuse	835,--
Aufpreis f.o. Systeme von 48K zu 64K	55,--
Aufpreis von 48K zu 64K+Z80 CPU	89,--
Apollo IIe (64K) vollbest. und geprüft	719,--
Apollo IIe (64K) + 80Z/24Z-Karte gepr.	759,--
Apollo IIe (128K) + 80Z/24Z-Karte geprüft	799,--
Apple IIe (64K) + 15er-Tastatur vollbestückt und geprüft	785,--
Apollo IIe ASKF (64 K) vollbestückt und geprüft	ab 876,--
Apollo IIe (64K) + Disk II F + Contr. +12" Monitor	1.387,--

APPLE IIe 128K+80Z/24Z Karte+Softw.	**1.998,--**
Disk-Drives voll APPLE II, 100%ig compatibel + R/W geprüft + Track ø Sensor:	
Disk//+Contr. + Kabel DOS 3.3 (Siemens) im Geh.	359,--
Disk//(Siemens) im Geh. + Kabel	**298,--**
Disk//+FHS (High Speed 1/2 Höhe) im Gehäuse + Contr. + Kabel	379,--
Disk//+HS (High Speed 1/2 Höhe) im Geh. + Kabel (Track Ac.<2mS)	**318,--**
Disk//+HS-80 Track im Gehäuse + Kabel f. orig. Contr. geeignet	389,--
Disk//+HS-80 w.o. + Contr. + DOS 3.3 + CP/M Patch-Softw.	489,--
Disk II-2x80 Track 640KB f. Erphi Controller modifiziert	299,--
Erphi DuoDisk 1,2 MByte im Gehäuse + Erphi-AFDC 3-Contr., gepr.	**898,--**

APPLESOFT + Tutorial + Reference Manual engl.	75,--
APPLE-DOS 3.3 Manual engl. = **25,---**/Hardcoh dt.	45,--
APPLE-Pascal Reference + Operating Manual engl.	48,--
APPLE-Fortran Manual engl.	69,--
CP/M-Softcard Vol. I + II, Manual engl.	69,--
Z80 Karte gepr. f. APPLE//c + RAM-Disk-Software + Hb.	447,--
Macintosh Umrüstung von 128K auf 512K	**335,--**
EPSON FX 85F/T+, 8bit/parallel 160 Z/S. NLQ + IBM Komp.	1.275,--
EPSON FX 105+, 8bit/parallel 15" sonst w.o.	1.585,--
EPSON Traktoraufsatz für FX 80 + FX 85	129,50
APPLE/EPSON-Drucker Graphic-Interface + Kabel	169,--
MX 80/82, FX/RX 80, FX 85 Spezialfarbband-Kassette	15,95
BROTHER-Typenraddr. HR 15XL 8bit/par. neuestes Modell	1.175,--

Wir führen verschiedene Monitore von Zenith, Phillips und Sanyo mit Video – TTL (IBM Komp.) oder Coloreingang. – Bitte Preisliste anfordern! –

Disketten in Box + Aufkleber I. Wahl, 10er Pack/100er Pack./ Stückpreis:

5¼"	BASF	3M SCOTCH	FUJI	DATA MAGNE-TICS	NEUTRAL
1X, SS/SD	3,09/ 2,89				
1D, SS/DD	3,29/ 2,99	3,38/ 2,98	4,27/ 3,97	1,96/1,86	1,85/1,75
2D, DS/DD	4,09/ 3,89	4,58/ 3,98	5,87/ 5,47	2,86/2,76	2,75/2,55
1D/96TPI	4,19/ 3,99	4,88/ 4,68			
2D/96TPI	4,79/ 4,59	5,48/ 5,18	7,87/ 7,67		
2HD/1,2MB	8,19/ 7,79	8,98/ 8,18	12,87/12,57		
1X, Hard	4,29/ 4,19				
8"					
1X, SS/SD	4,49/ 4,39	6,68/ 6,48	6,97/ 6,67		
1D, SS/DD	4,79/ 4,59	6,58/ 6,28			
2D, DS/DD	6,49/ 5,99	9,88/ 9,58	10,17/ 9,77		
1X, Hard	**4,49/ 4,39**	**6,68/ 6,48**			
3½"					
Einseitig	5,89/ 5,59	6,78/ 6,08	7,67/ 7,17		
Zweiseitig	7,39/ 6,99	9,28/ 8,48	11,17/ 9,97		

Das sind Preise!

5¼" Disketten-Archivbox für 10 Disks (1 St./10 St.)	4,75/43,50
5¼" Disk-Karteikasten Kunststoff (ca. 80 Disk)	26,50
5¼" Disk-Karteikasten Kunststoff (ca. 80 Disk) Rauchglas	45,95
5¼" Disk-K.K.Rauchglas, abschließb. (f. ca. 80-100 Disks)	47,50
2000 Bl. Tabpapier (24 cm x 12" einf.) weiß oder grün/weiß perf.	45,--
4000 Et.-Aufkl. doppelr. (107 x 36 mm auf 240 x 12" perf. Trägerp.)	66,50
4000 Etik.-Aufkl. einreih. (107 x 36 mm auf 125 x 12" perf. Träg.)	65,--

OSZILLOSCOPE HAMEG ab Lager

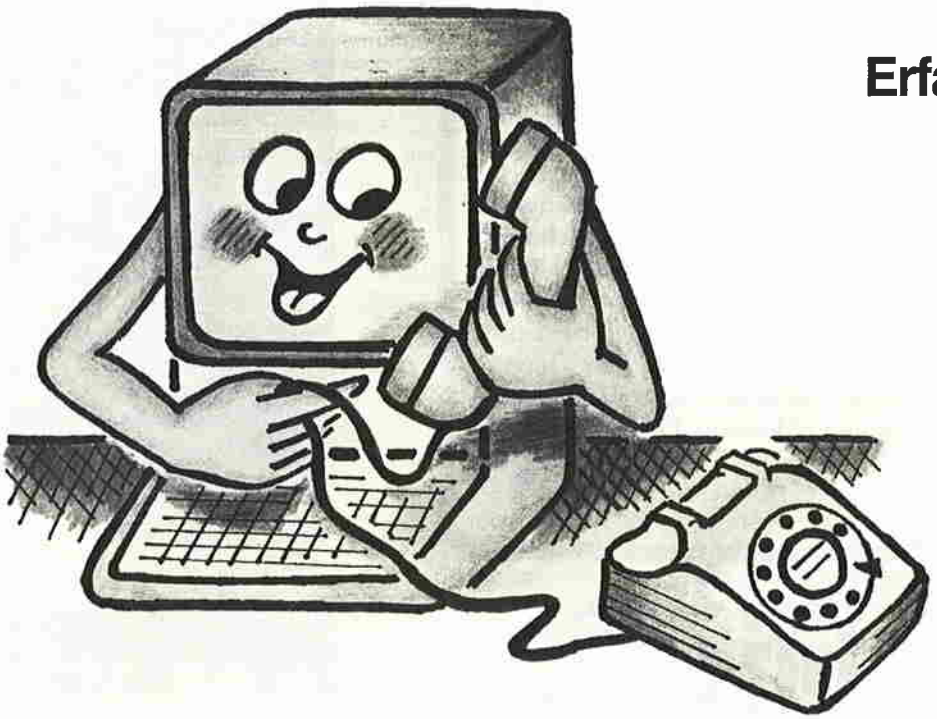
Bei Vorauszahlung frei Empfangsstation unversichert in der BRD, ausgenommen Papier und Etiketten, sonst N.N. + V.S. ab DM 30,--

Öffnungszeiten: Mo, Di, Do, Fr v. 10-18 h, Mi u. Sa v. 10-14 h, Ia Sa v. 10-18 h.
Telef. Best.: Mo, Di, Do, Fr von 10-19 h, Mi u. Sa wie Öffnungszeiten.
GEWÄHRLEISTUNG: 6 Monate, auf alle bei uns gekauften Geräte, durch unsere eigene Service-Werkstatt.

REPARATUREN an Apple + compatiblen Geräten + Zubehör führt unser Spezialistenteam garantiert zuverlässig + besonders kostengünstig aus. **Sprechen sie mit uns. Kostenvorschlag auf Wunsch!**

aaa electronic gmbh
Telex 0772642 aaa-d
Habsburgerstraße 134
7800 FREIBURG, Tel (0761) 276864
Bauelemente – Bausätze – µP's
Meßgerate – Zubehor – Fachliteratur
Fachgeschäft fur Elektronik + Mikrocomputer

Hallo, hier spricht der Apple



Erfahrungen mit einem Akustikkoppler

von Franz-Josef Hüskens

1. Ausgangsposition

Mittlerweile sind Mikrocomputer weit verbreitet und bieten eine Vielzahl von Anwendungsmöglichkeiten im professionellen und privaten Bereich. Für viele Computerbesitzer ist es aber schon nicht mehr interessant genug, nur zu Hause im stillen Kämmerlein abgeschnitten von der Umwelt mit ihrem Rechner zu „spielen". Viele Anwender wollen Daten oder Programme von anderen Orten bekommen oder dorthin weitergeben. Dabei sollen diese Daten schnell und sicher – nach Möglichkeit ohne Verwendung von „harten" Datenträgern wie Disketten oder Papier – transportiert werden. In den letzten Jahren hat daher die Kommunikation von Computer zu Computer immer mehr an Bedeutung gewonnen. Im kommerziellen Bereich war und ist die Kommunikation zwischen Rechnern bereits eine Selbstverständlichkeit. Die ziemlich teure Anmietung eines Modems von der Deutschen Bundespost schreckte in der Vergangenheit viele Interessenten ab (1983 kostete dies noch DM 80,– im Monat zuzüglich der normalen Fernmeldegebühren, 1986 monatlich DM 50,– und DM 65,– für den Anschluß). Aber auch die Post erkannte wohl die Zeichen der Zeit und ermöglicht nun auch „kleinen" Anwendern die Datenfernübertragung. So erhalten immer mehr Akustikkoppler die amtliche Zulassung der Bundespost. Akustikkoppler sind Modems, die akustisch an eine Telefonleitung angekoppelt werden, d.h. der Telefonhörer wird so auf das Modem gelegt, daß dieses Töne empfangen und senden kann. Akustikkoppler gibt es (wie fast alles) in jeder Preislage. Da eine solche Anschaffung für die Hobbyisten auch zu einer Preisfrage wird, soll hier nun eines der preiswerteren Modems dieser Art vorgestellt werden. Der Akustikkoppler heißt „dataphon s21d", hat die FTZ-Zulassungsnummer FTZ 18.13.1917.00 und wird von der Firma Daum Electronic in Veitsbronn hergestellt. Er arbeitet nach der Vorschrift CCITT V.21 mit einer Übertragungsgeschwindigkeit bis zu 300 Bit/s. Vertrieben wird der Koppler u.a. von der Firma Hib in Nürnberg, die nicht nur die passende Hardware zum An-

schluß des Kopplers an den Game-Port des Apple, sondern auch ein Terminalprogramm mitliefert. Ein solches Programm ist genauso wichtig wie der Akustikkoppler selbst, da ohne diese Software die empfangenen Daten nicht aufbereitet und für den Benutzer in verständliche Zeichen umgewandelt werden können. Außerdem wäre ohne Terminalprogramm auch kein Dialogbetrieb mit dem angewählten Rechner möglich.

2. Das dataphon

Dieser Akustikkoppler kann mit einer 9V-Blockbatterie, einem Akkumulator oder einem Netzgerät betrieben werden. Ebenso soll auch der Betrieb über den Schnittstellenstecker möglich sein. Die Verwendungsmöglichkeit von Batterie bzw. Akku ermöglicht zusammen mit tragbaren Computern (für den Apple IIc jedoch bisher nur in Verbindung mit einer tragbaren Stromerzeugungsanlage) auch einen mobilen Einsatz, z.B. von öffentlichen Telefonzellen aus. Bei der Benutzung mit Batterien beträgt die Betriebsdauer nach Angaben des Herstellers 8 Stunden, mit dem Akku 4 Stunden. Die Elektronik des Geräts sitzt in einem grauen Kunststoffgehäuse. Dieses Gehäuse hat in der Mitte einen biegsamen, „ziehharmonikaartigen" Teil. Damit ist es möglich, Telefonhörer der unterschiedlichsten Bauarten zu benutzen. Die Hör- und Sprechmuscheln des Telefonhörers werden in die Schaumstoffringe des Akustikkopplers gepreßt. Die eckigen Muscheln der neueren Telefonmodelle rutschen jedoch aus den Schaumstoffringen heraus. Hier empfiehlt es sich, Telefonhörer und Akustikkoppler auf die Seite zu legen, wobei der Handapparat gewichtsmäßig stärker belastet werden muß.

2.1. Bedienung

An einer Längsseite des Akustikkopplers befinden sich zwei Schalter: der Ein-/Aus-Schiebeschalter und der Schiebeschalter zur Auswahl des benutzten Frequenzpaares. Letzterer besitzt drei verschiedene Stellungen: Die Stellung ANS (answer-mode) wird gewählt, wenn angerufen wurde, die Stellung ORIG (originate-mode) steht für eigene Anrufe, die Stellung AUTO (automatic) dient der automatischen Kanalwahl. Der Frequenzschalter sollte, wenn man eine Mailbox oder eine Datenbank anruft, auf ORIG stehen. Ruft man einen Bekannten an, der ebenfalls das „dataphon" angeschlossen hat, so muß dieser sein Gerät auf ANS oder AUTO einstellen. Neben den beiden Schaltern sitzen drei Leuchtdioden, die im Betrieb durch Blinken den jeweiligen Modus anzeigen bzw. die Sendebereitschaft signalisieren. Die 25polige Schnittstellenbuchse und der Anschluß für ein externes Netzteil sitzen an der schmaleren Stirnseite des Kopplers.

2.2. Begleitheft

Mitgeliefert wird ein 38 Seiten starkes Heft, in dem einige Grundlagen der Datenübertragungstechnik vermittelt und die Benutzung des Akustikkopplers erklärt werden. Man erhält Informationen über die Umwandlung von Computerdaten, die eine Übertragung über das Fernsprechnetz erst ermöglicht, über die Normung von Signalfrequenzen, über die Schnittstelle zum Computer und über das Übertragungsformat. Dann folgt die Beschreibung des „dataphons" und seiner Inbetriebnahme. In der Bedienungsanleitung wird kurz erklärt, wie man den Koppler an den Rechner anschließt und wie der Verbindungsaufbau erfolgt. Der Betriebsablauf in den drei verschiedenen Modi wird erläutert, und schließlich werden in der Anleitung noch allgemeine Hinweise zum Betrieb des „dataphons" gegeben. Im Anhang werden neben dem Literaturverzeichnis einige Rufnummern öffentlich zugänglicher Datenzentralen aufgeführt. Zusätzlich enthielt mein „Kommunikationspaket" noch das „Mailbox-Telefonbuch", einen Sonderdruck des DFÜ-Anbieterverzeichnisses, das von Claus Warneke in Bremen herausgegeben wird. Dieses Buch enthält auf über 100 Seiten alphabetisch sortiert – von A (wie Aachen) bis W (Wuppertal) – ein Verzeichnis der verschiedensten Mailboxen mit Angaben der Rufnummern, der Betreiber und weiterer wichtiger Informationen. Dazu gehören u.a. der Name der Box, die erforderliche Parametereinstellung, das Angebot der Mailbox und ihre Betriebszeit. Eine Seite des Telefonbuchs enthält die Rufnummern einiger Mailboxen aus Übersee (Amerika, Australien usw.), außerdem werden Rufnummern angegeben, die nur über Datex-P-Verbindungen erreicht werden können.

3. Das Terminalprogramm

Wie anfangs erwähnt, ist ein einfach zu benutzendes Terminalprogramm ebenso wichtig wie der Akustikkoppler bzw. Modem selbst. Mitgeliefert wurde bei meinem Paket das Kommunikationsprogramm „Hib Modem-Transfer" in der Version 331-M und das Schnittstellenkabel. Das Programm ermöglicht Datenfernübertragung über den Game-Port des Apple. Es ist laut Herstellerangaben für jeden Akustikkoppler mit V.24-Schnittstelle geeignet. Da die Datenübertragung über den Game-Port vorgenommen wird, bedeutet dies für den Apple, daß er nicht nur fertige Daten empfängt und eigene Daten weitergibt, vielmehr muß der Computer hier die gesamte Übertragung bewerkstelligen.

3.1. Anwendung

Die Mini-Bedienungsanleitung für das Terminalprogramm ist mit 8 Seiten zwar knapp gehalten, aber voll ausreichend. Zuerst wird der Anschluß des Schnittstellenkabels an den Apple erklärt, danach folgt die kurze, aber gehaltvolle Beschreibung des Programms. Nach dem Start zeigt eine mit dramatischer Musik unterlegte Grafik, welche Firma das Programm vertreibt. Danach erscheint das Hauptmenü, das die Grundfunktionen *Transfer, Laden, Speichern, Tastaturkonfiguration, Ende* zur Auswahl anbietet. Zusätzlich kann man im Hauptmenü verschiedene Parameter einstellen.

Zur Auswahl einer Grundfunktion bzw. zur Neueinstellung eines oder mehrerer Parameter bewegt man den Cursor mit den Tasten I, J, K oder M (bzw. A, ←, →, Z) auf die gewünschte Stelle. Bei den Parametern wird jeweils die momentane Einstellung angezeigt und – falls der betreffende Parameter durch Positionierung des Cursors angewählt wurde – durch Druck auf die Return-Taste um eine mögliche Einstellung weitergeschaltet. Die Anzahl der Daten- und Stoppbits kann geändert und ein Paritätsbit gesendet werden. Empfangsseitig wird das Paritätsbit jedoch vom Terminalprogramm ignoriert.

Mit der Auswahl eines Filters kann der Benutzer angeben, ob er Code (Programme usw.) oder Text empfangen will. Mit einem weiteren Parameter gibt er an, ob der automatische Zeilenvorschub nach jedem Return an- oder ausgeschaltet sein soll. Bei der Anwahl des Parameters „Drucker" wird in der Grundfunktion „Transfer" jeder Eingang gedruckt. Außerdem wird der vor dem Speichern wahlweise mögliche Kontrollausdruck über den Drucker ausgegeben. Die XON/XOFF-Funktion erlaubt dem angekoppelten Rechner, eine automatische Übertragung an ihn mit einem beliebigen Zeichen zu stoppen und wieder zu starten. Zusätzlich kann die Dateiform bestimmt werden, die eine Datei beim Speichern auf bzw. Laden von Diskette haben soll. Hier wird zwischen vier Dateitypen unterschieden:

– TEXTdateien: Sie sollten normalerweise verwendet werden, da hier eine leichte Fehlererkennung möglich ist.

– APPLESOFTdateien: Wegen möglicher Übertragungsfehler (Leitungsknacken usw.) sollte diese Dateiform nur bei weniger wichtigen Programmen verwendet werden.

– BINÄRdateien: Manche Textverarbeitungsprogramme oder andere Programmiersprachen wie Logo oder Forth speichern Texte als Binärfiles. Sie können – wie HGR-Seiten auch – mit den obigen Einschränkungen übertragen werden.
– SPEZIALdateien: Sie dienen zum Laden und Speichern von Tastaturdefinitionen (s.u.).

3.2. Grundfunktionen des Hauptmenüs

Zu den Grundfunktionen des Hauptmenüs gehören die vier Funktionen, die im folgenden näher erläutert werden.

3.2.1. Tastaturdefinition

Hiermit kann man die (wie es in der Bedienungsanleitung so schön heißt) „Tastaturausgangsübersetzungstabelle" nach eigenen Wünschen zusammenstellen. Die neue Tabelle wird wahlweise in einer von zwei Speicherbereichen (Bank) abgelegt. Die Bank wird mit Ctrl-L ausgewählt und bleibt dann bis zum nächsten Ctrl-L eingestellt. Hiermit wird es möglich, Zeichen, die man normalerweise nicht eingeben kann, frei zu definieren und auf Wunsch abzurufen. Um verschiedene Tastaturbelegungen nicht immer neu von Hand eingeben zu müssen, können die eigenen Tabellen auf Diskette gespeichert werden.

3.2.2. Speichern

Dieser Programmpunkt wird benötigt, um den Speicherinhalt wahlweise auf dem Drucker oder Bildschirm auszugeben und um den Speicherinhalt auf Diskette zu schreiben.
Das Terminalprogramm kann allerdings keine bestehenden Dateien überschreiben.

3.2.3. Laden

Mit dieser Menüoption können vorbereitete Texte von Diskette in den Speicher geladen werden. Dabei wird der bestehende Speicherinhalt gelöscht; das Programm macht vor Beginn des Ladevorgangs jedoch auf diese Tatsache aufmerksam. Ist eine Datei so groß, daß sie nicht in den Speicher paßt, wird ein entsprechender Hinweis gegeben und nur ein Teil geladen.

3.2.4. Transfer

Mit diesem Programmteil wird die Datenübertragung durchgeführt. Der Bildschirm zeigt eine Statuszeile, das Eingangsfenster und das Ausgangsfenster.
Die *Statuszeile* zeigt dem Benutzer, was gerade im Speicher vor sich geht. Mit verschiedenen Zeichen wird gezeigt, daß

– der Speicher leer ist
– der Speicher im Moment „mitschreibt"
– Daten im Speicher sind und noch weitere Daten aufgenommen werden können
– der Speicher „mitschreibt", aber bald voll ist
– der Speicher voll ist; weiteres „Mitschreiben" ist nun erst nach Löschen des Speichers wieder möglich.

Daneben gibt es weitere Symbole, die verschiedene Sachverhalte anzeigen, z.B. daß eine Datei mit der Funktion „Laden" in den Speicher gelegt wurde und nach dem Anschalten des Speichers (s.u.) gesendet wird. Zuvor können noch Texte direkt eingegeben und abgesandt werden. Ein anderes Symbol verdeutlicht, daß momentan Daten übertragen werden. Dabei kann die Übertragung reell nur mit aktivierter XON/XOFF-Funktion durch die Gegenstation unterbrochen werden. (Drücken der Reset-Taste unterbricht auch, ist aber hier nicht angebracht!) Eine derartige Unterbrechung der Datenübertragung wird mit einem gesonderten Zeichen dargestellt. Solange die Gegenstation kein XON-Signal sendet, können untereinander zusätzliche Informationen ausgetauscht werden.
Im *Eingangsfenster* werden alle Zeichen angezeigt, die vom Rechner empfangen werden.
Das *Ausgangsfenster* protokolliert zu Kontrollzwecken alle abgesandten Zeichen.
Im Transfermodus sind folgende Eingaben zur Beeinflußung der Übertragung möglich:
1. ESC: Verlassen des Transfermodus und Rückkehr in das Hauptmenü
2. CTRL-§: Bildschirm (Eingangs- und Ausgangsfenster) löschen
3. CTRL-A: Speicher anschalten (Mitschreiben bzw. Senden)
4. CTRL-E: Speicher ausschalten
5. CTRL-B: Speicher löschen
6. CTRL-L: Auswahl der anderen „Tastaturausgangsübersetzungstabelle"

4. Der Verbindungsaufbau

Will man mit dem beschriebenen Kommunikationspaket Kontakt mit einer Mailbox oder Datenbank aufnehmen, müssen als erstes die Übertragungsparameter richtig eingestellt werden. Dann kann man versuchen, die Fernsprechverbindung zu der Mailbox aufzubauen. Hier treten oft schon die ersten Schwierigkeiten auf: Die meisten öffentlich (dies bedeutet auch ohne Gebühren) zugänglichen Mailboxen sind öfter besetzt. Auch das Ausweichen in die Abend- oder sogar Nachtstunden (für die Telefonrechnung von Vorteil) bringt hierbei selten eine Entspannung der Situation, da offenbar alle Mailbox-Benutzer Nachtmenschen sind. Man muß häufig viel Geduld und Ausdauer haben, bis der gewünschte Anschluß endlich frei ist. Die am stärksten frequentierten Mailboxen haben daher eine „Time-Out-Funktion" eingebaut und lassen nur eine zeitlich begrenzte Nutzung zu. Bei Zeitüberschreitung wird – nach einem entsprechenden Hinweis – die Verbindung aufgelöst.
Hat man jedoch das Glück, einen „freien" Anschluß zu erwischen, hört man nach dem Freizeichen einen andauernden Piepston. Spätestens jetzt muß der Telefonhörer in die Muscheln des Akustikkopplers gelegt werden. Ein Druck auf die Returntaste läßt dann wie von Zauberhand die Begrüßung der Mailbox auf dem Bildschirm erscheinen. In einem mehr oder weniger umfangreichen Menü stehen dem Anrufer verschiedene Auswahlpunkte zur Verfügung. Diese reichen von Informationen und – falls man eine kommerzielle Box angerufen hat – Werbung über kleine Programme usw. (also Angebote des Betreibers) bis hin zu Kontaktgesuchen, Nachrichten und weiteren Informationen anderer Mailboxbenutzer. Da die Mailboxen benutzerfreundlich aufgebaut sein sollten, bietet das Hauptmenü für weniger versierte Benutzer häufig zusätzlich eine Hilfsfunktion. Darin wird erklärt, wie man mit der Mailbox arbeitet. Sollte man auch damit nicht klarkommen, kann man eventuell Hilfe durch den „SysOp" (System-Operator = Mailbox-Betreuer) erhalten. Der kann nämlich feststellen, ob der Benutzer Hilfe braucht und dann mit gezielten Fragen (im Dialog) Probleme lösen helfen. Einige Mailboxen gestatten es sogar, mit einem eigenen Menüpunkt den SysOp zu rufen. Voraussetzung für die Kontaktaufnahme ist natürlich, daß man während der normalen Arbeitszeit des SysOps anruft.

Fazit

Zusammenfassend kann man sagen, daß sowohl die Hardware (der Akustikkoppler) als auch die Software (das Terminalprogramm) einfach zu bedienen sind. Der einzige Nachteil, den dieses (aber auch jedes andere) „Kommunikationspaket" mit sich bringt, macht sich meist erst nach einiger Zeit bemerkbar, nämlich dann, wenn die Telefonrechnung kommt!
Nach den ersten „Übungen" mit dem Akustikkoppler und dem Terminalprogramm mußte ich mich doch sehr zurückhalten, nicht öfter auf Entdeckungsreise in die mir bekannten Mailboxen zu gehen. Allerdings macht es auch bei relativ seltener Benutzung des Pakets noch viel Spaß, dem Apple die große Welt zu zeigen.

Bücher

Logo in 100 Beispielen

von K.Menzel
1985, 234 S., kart., DM 23,80
Teubner Verlag, Stuttgart
Gliederung
Einleitung – Aufbau eines Logo-Programms – Elemente von Logo – Datenbehandlung in Logo – 100 Logo-Beispiele – Logo-Latein
Bemerkungen
Mit dieser Beispielsammlung kann der Einsteiger die Haupteigenschaften von Logo und den Grundaufbau der Logoprogramme erlernen. Im fünften Kapitel werden 100 Beispielprogramme in einheitlicher Form wiedergegeben und beschrieben (alle 100 Beispiele sind auch im entsprechenden Teubner-BASIC-Band aufgeführt). Die beiden Programmiersprachen BASIC und Logo werden im letzten Kapitel einander gegenübergestellt; der Benutzer eines Heim- oder Personalcomputers hat selbst die Möglichkeit, die Eigenschaften von Logo und BASIC miteinander zu vergleichen und sich ein eigenes Urteil zu bilden.

CP/M und MS-DOS Mikrocomputer

von Ekkehard Kaier
1984, 234 S., kart., DM 48,–
Vieweg Verlag,Braunschweig
Gliederung
Computer – Hardware – Software – Firmware – MBASIC auf Mikrocomputern: Erstes Anwenderprogramm – Kurzbeschreibung von MBASIC – MBASIC auf IBM-PC, Sirius 1, Olympia People, Alphatronic PC – Programmierung in MBASIC: Grundlegende Programmstrukturen an Beispielen – Textverarbeitung – Ein- und Ausgabe – Maschinennahe Programmierung – Programme überprüfen und verbinden – Tabellenverarbeitung – Grafik – Suchen, Sortieren, Mischen und Gruppieren von Daten – Sequentielle, Direktzugriff-, Indexsequentielle, Verkettete Dateien – Gekettete Liste – Binärer Baum
Bemerkungen
Dieser Band der „Wegweiser-Reihe" versteht sich als Ergänzung zu System-Handbüchern und beschreibt die Grundlagen der DV, um sie an Anwendungsbeispielen für den Mikrocomputer zu veranschaulichen. Das Buch enthält einen kompletten Programmierkurs in MBASIC mit 86 Beispielprogrammen, die in codierter Form und als Ausführung (Run) wiedergegeben und beschrieben werden. Inhaltsverzeichnis und Sachwortregister sind ausführlich und zum Nachschlagen geeignet. Die Programme und Dateien des Buches können für DM 68,– als Diskette bezogen werden (unter CP/M 2.20 für Apple IIe und Kompatible und PC-DOS/MS-DOS für IBM und Kompatible).

CBASIC

Das Einführungs- und Nachschlagebuch für Anwender
von W. Eggerichs und R. Weiß
1985, 172 S., kart., DM 39,80
Dr. Alfred Hüthig Verlag, Heidelberg
Gliederung
Allgemeines – Erstellen und Durchführen von CBASIC-Programmen – Datentypen, Variablenbenennung, Ausdrücke und numerische Funktionen – Datenein- und -ausgabe – Allgemeine CBASIC-Anweisungen – Entscheidungs-, Wiederholungs- und Feldbefehle – Funktionen und Unterprogramme – Formatierte Ausgaben und Bildschirmsteuerung – Dateiorganisation – Beispielprogramm zur Dateiverwaltung – Weitere CBASIC-Kommandos in Kurzform – Lösungen zu den Übungsaufgaben – Das Interpretersystem CBAS und CRUN – Das Compilersystem CB86 und LINK86
Bemerkungen
Die Programmiersprache CBASIC ist eine kaufmännisch, verwaltungstechnisch orientierte Variante von BASIC und läuft unter den Betriebssystemen der CP/M-Familie. Die Vorteile von CBASIC liegen in der hohen Rechengenauigkeit, der Möglichkeit zur strukturierten (und zur selbstdokumentierenden) Programmierung und zur Erzeugung von schnell lauffähigen, compilierten Programmen. In diesem Buch werden die CBASIC-Anweisungen erläutert und Unterschiede zwischen Interpreter- und Compilerversion aufgezeigt. Der Leser wird anhand zahlreicher Beispiele an die Anwendung von CBASIC herangeführt und erhält zusätzlich eine Einführung in den CP/M-Editor.

Multiplan-Training auf Apple IIe und Apple IIc

Eine leicht lesbare systematische Einführung
von B. Kretschmer
1985, 257 S., kart., ca. DM 35,–
Vogel Verlag, Würzburg
Gliederung
Einleitung – Was Sie vor der ersten Aufgabe wissen und tun sollten – Einfache Multiplan-Tabelle – Tabellengestaltung – Ändern des Tabelleninhalts – Mehrdimensionale Daten – Rechen- und logische Funktionen – Textverarbeitung – Dateiverarbeitung – Arbeiten mit mehreren Dateien
Bemerkungen
Multiplan ist ein sehr leicht handhabbares, professionelles Tabellenkalkulationsprogramm, mit dem man Rechenaufgaben, Texte und kleine Dateien bearbeiten kann. Das Buch zeigt an einfachen Beispielen die Anwendung von Multiplan bei Umsatzstatistiken, Textverarbeitung, Serienbrieferstellung, Erstellen, Pflegen und Verwenden von Dateien, Lieferscheinschreibung etc. Alle erforderlichen Eingaben am Apple IIc/IIe und alle Bildschirminhalte und Ausdrucke sind originalgetreu wiedergegeben. Dem Benutzer unter Betriebssystem CP/M dienen Installationshinweise und eine Übersicht über die Unterschiede bei der Belegung der Funktionstasten.

Applesoft-BASIC total

Programmiertechniken und Anwendungen für Apple II+, IIe und IIc
von J.Robra
1985, 342 S., kart., DM 45,–
Vogel Verlag, Würzburg
Gliederung
Grundlagen moderner Programmiertechnik – Der BASIC-Apple: Betriebssystem DOS 3.3 – Applesoft-BASIC – Applesoft-Programme – DOS-Dateien – Grafik – Anwendungsbeispiele: Textverarbeitung – Business Graphics – Schlußbemerkungen – Anhang
Bemerkungen
Dieses Buch wendet sich an alle, die für Apple II+, IIe und IIc mit Applesoft-BASIC unter DOS 3.3 leistungsfähige Programme aufstellen wollen. Es zeigt beispielhaft, wie aus einem Problem ein sauber aufgebautes und strukturiertes, benutzerfreundliches Programm entsteht. Nach einem Überblick über den Aufbau von DOS 3.3 und einer nach Funktionen gegliederten und mit Beispielen belegten Darstellung aller Befehle und Möglichkeiten von Applesoft-BASIC und DOS 3.3 werden aufwendige Programme aus mehreren Sachgebieten einschließlich Grafik schrittweise bis zur Einsatzbereitschaft entwickelt.

Maschinensprache? – Kein Problem

Grundlagen und Programmierung der Mikroprozessoren 6502/6510 für Anfänger
von M. Stenzel
1986, 264 S., kart., DM 58,–
Franzis Verlag, München
Gliederung
Die Elemente eines Mikrocomputersystems – Der organisatorische Aufbau des Hauptspeichers – Umrechnung der verschiedenen Zahlensysteme – Der innere Aufbau der CPU – Maschinensprache – was ist das eigentlich? – Der Befehlssatz des 6502 – Wie man ein Maschinenprogramm zu Papier und in den Rechner bringt – Vom Zählen – Das Status-Register – Das Indizieren – Rechnen mit dem 6502 – Datenverarbeitung auf logischer Basis – Suchen und Finden – Zeiger und Vektoren – Der Stapel – Entwickeln eines größeren Maschinenprogramms – Der Viertelpunktplot bei Commodore-Rechnern – Zeitfragen – Reset und Interrupt – Ausblick – Anhang
Bemerkungen
Im Gegensatz zu rein systematischen Darstellungen wird die Maschinensprache in diesem Buch auch wirklich als Sprache gelehrt. Die Anleitungen sind nicht auf einen bestimmten Computertyp festgelegt, sondern treffen für alle Computer mit einem Prozessor aus der 6502-Familie zu. Im Mittelpunkt steht die Programmierpraxis. Zunächst werden Grundkenntnisse vermittelt, danach lernt der Anwender stufenweise neue Vokabeln, Grammatikteile und Programmiertricks. Nach kurzer Zeit kann er erfolgreich eigene Programme entwerfen oder fremde Programme für die eigenen Aufgaben abändern.

Die Hackerbibel

Hrsg. Chaos-Computer-Club, Hamburg
1985, 260 S., kart., DM 33,33
Verlag Grüne Kraft – W. Pieper, Löhrbach
Bemerkungen
Die Hackerbibel ist von Amateuren im Stil einer Zeitschrift aufgemacht und enthält eine (unübersichtliche) Fülle englisch- und deutschsprachiger Berichte zum Thema Perspektiven, Chancen und Gefahren der Fern- und Nahkommunikation. Passend zum Namen des Herausgebers ist die Hackerbibel wohl von Chaoten für Chaoten produziert, die den Durchblick über das „Inhaltshickhack" (Zitat) bewahren können.

dBase II

Band 2 Programmierung
von W. Eggerichs
1986, 2. Auflage, 191 S., kart., DM 39,80
Dr. Alfred Hüthig Verlag, Heidelberg
Gliederung
Datenbanken, Datentypen, Variablen, Ausdrücke und Kommandoerstellung – Befehle für Kommandodateien – Arbeiten mit Speichervariablen – Logische Entscheidungen – Weitere Funktionen in dBase II – Die SET-Kommandos – Bearbeitung einer vollständigen Programmieraufgabe – Ergänzende Programme und Systeme – Kurzbeschreibung aller Kommandos und Funktionen – Ausblick – Besprechung der Übungsaufgaben
Bemerkungen
Die dreibändige Buchreihe befaßt sich mit dem Datenbanksystem dBase II, das speziell für Mikrocomputer entwickelt wurde und unter den Betriebssystemen CP/M, MP/M, MS-DOS und PC-DOS läuft. In den ersten beiden Bänden werden Kommandos bestimmter Leistungsbereiche herausgefiltert, erläutert und der Kenntnisstand des Lesers mit kleinen Aufgaben überprüft.
Der zweite Band soll dem Leser den Schritt von der reinen Dialogarbeit zu einer verstärkten Arbeit mit den Kommandodateien (den dBase-II-Programmen) zeigen. Dabei geht das Buch vorwiegend auf die Problemanalyse bei der Programmentwicklung ein. Entsprechend umfangreich stellt sich die dBase-II-unabhängige Erarbeitung und Aufarbeitung der Programmlogik dar.

50 Pascal-Programme

von B. H. Hunter
1985, 320 S., kart., DM 48,–
Sybex-Verlag, Düsseldorf
Gliederung
Einfache Programme – Geschäftsprogramme – Finanzprogramme – Programme für den persönlichen Gebrauch – Hilfsprogramme – Mathematikprogramme – Programme für den Maschinenbau – Elektrizitätsprogramme – Ingenieurprogramme – Spiele
Bemerkungen
Diese Programmsammlung wendet sich an kommerzielle und private Anwender mit Grundkenntnissen in Pascal. 50 nützliche Programme aus dem mathematischen, technisch-naturwissenschaftlichen, wirtschaftlichen und privaten Bereich und deren Einsatzmöglichkeiten werden vorgestellt. Der Leser kann einerseits die aufgelisteten Programme direkt nutzen; zum andern ermöglichen ihm die klare Struktur und ausführliche Kommentare eine Anpassung der Programme auf seine individuellen Probleme und Fragestellungen.

Pascal in Übungsaufgaben

Fragen, Fallen, Fehlerquellen
Unter besonderer Berücksichtigung von Turbo-Pascal und UCSD-Pascal
von W. J. Weber
1986, 152 S., kart., DM 23,80
Teubner Verlag, Stuttgart
Gliederung
Einleitung – Grundlagen – Arithmetik – Boolesche Ausdrücke und Variable – Schleifen – Datentypen – Fehler und Fehlerquellen – Unterprogramme – Dateien – Dynamische Datenstrukturen – Programmierstil – Methodische Überlegungen
Bemerkungen
Die in diesem Buch gesammelten Aufgaben und Probleme behandeln wichtige Haupteigenschaften von Pascal, aber auch unbekanntere Eigenarten, „Fallen". Die Aufgabensammlung wurde zwei Jahre lang in der Unterrichtspraxis erprobt und verbessert und berücksichtigt häufig auftretende Lern- und Verständnisschwierigkeiten. Der Umgang mit den Daten- und Kontrollstrukturen von Pascal hat Vorgang vor dem „allgemeinen Programmieren" anderer Aufgabensammlungen. Zu allen Aufgaben werden kommentierte Lösungen gegeben, so daß das Buch auch zum Selbststudium und zur Prüfungsvorbereitung geeignet ist. Die Thematik der einzelnen Aufgabenbereiche wird nur kurz in Form eines Überblicks am Kapitelanfang erläutert, deshalb sollte der Anwender parallel zur Sammlung ein Pascal-Lehrbuch benutzen.

Grundlagen der Programmiersprache PL/1

von G. Smolek und M. Weissenböck
1985, 150 S., kart., DM 42,–
Dr. Alfred Hüthig Verlag, Heidelberg
Gliederung
Algorithmen und Struktogramme – Lese- und Schreibbefehl, Namen und Zeichenketten – Einfache mathematische Funktionen – Auswahl- und Wiederholungsanweisungen – Weitere Funktionen, Bereiche – Weitere Lese- und Druckbefehle, Formatangaben, dynamische Felder – Files, Blöcke, ON-Anweisungen – Blockstruktur, Unterprogramme – Rekursion, externe Funktionen, Funktionsprozeduren – Fehler und Testhilfen – Marken, Sprungbefehle, Satzweise Ein- und Ausgabe – Jobsteuerung, Outputbeschreibung – Lösungen zu den Übungsaufgaben
Bemerkungen
In diesem Buch werden die Grundgedanken des Programmierens allgemeingültig und unter besonderer Beachtung der strukturierten Programmierung am Beispiel von PL/1 behandelt. Der Aufbau der Programme wird anhand von verständlichen Beispielen erläutert und die damit verbundenen EDV-Probleme erklärt. Am Beginn jedes Kapitels befindet sich ein kurzer Inhaltsüberblick, am Ende folgen Übungsaufgaben. Zusatzergänzungen sind in kleiner Schrift gehalten.

Assembler-Programme zum Apple-II

Der problemlose Aufstieg von BASIC in die Assemblersprache
von E.Esders
1986, 204 S., kart., DM 48,–
Franzis Verlag, München
Gliederung
Der Applesoft-BASIC-Interpreter – Erweiterungen des Applesoft-BASIC – HIRES-Fenster HRWSR – HIRES-Zeichengenerator
Bemerkungen
Dem Apple-Anwender bietet sich sowohl eine umfassende Einführung in das für ihn neue Gebiet der Assemblersprache als auch eine äußert wertvolle Sammlung von aussagekräftigen und nützlichen Programmen. Diese Programme bilden den Kern des Buches. Sie lassen sich als fertige Utilities ebenso anwenden wie als Bausteine in Verbindung mit BASIC- oder Pascal-Programmen.

PEEKER Börse

Gelegenheitsanzeigen

Sie können unter dieser Rubrik zu einem besonders günstigen Preis

- Ihre Hardware und Software verkaufen
- Ihre Hard- und Software suchen
- Kontakte knüpfen und vieles mehr

Musteranzeige privat

1 Druckzeile à 32 Buchstaben nur DM 5,50 zuzügl. ges. MwSt. Mindestens 2 Druckzeilen
Beispiel:

> Verkaufe neuwertigen Typenraddrucker mit Apple-Interface. Preis auf Anfrage. Tel. 007

nur DM 18,81 inkl. MwSt.

Musteranzeige gewerblich

1 Druckzeile à 32 Buchstaben nur DM 11,– zuzügl. ges. MwSt. Mindestens 2 Druckzeilen
Beispiel:

> Neu im Angebot: Professionelle, separate Tastatur für Apple II plus 16 Funktionstasten und separatem Ziffernblock.
> Fa. Keyboard & Co.

nur DM 62,70 inkl. MwSt.

PEEKER

Peeker-Börse

AUFTRAG FÜR KLEINANZEIGEN

Bitte veröffentlichen Sie in der nächsterreichbaren Ausgabe nachstehenden Text unter folgender Rubrik:

☐ suche Hardware ☐ suche Software ☐ Verschiedenes
☐ biete Hardware ☐ biete Software ☐ Chiffre

Bitte jeweils 32 Buchstaben pro Zeile – einschließlich Satzzeichen und Wortzwischenräume. Bitte Absender nicht vergessen. Mindestens 2 Zeilen. Chiffregebühr DM 6,– zuzügl. MwSt.

PEEKER

Produkt-Karte

Zu der im **Peeker**, Heft _____, Seite _____ erschienenen

Anzeige über _____

_____

bitte ich um detaillierte Information.

Ich wünsche ☐ Prospekt, Datenblatt ☐ Preisliste ☐ schriftliches Angebot ☐ tel. Rückruf

Menge	Produkt und Bestellnummer	à DM	gesamt DM

gebe ich nebenstehende Bestellung unter Anerkennung Ihrer in der Anzeige genannten Liefer- und Zahlungsbedingungen auf.

_____
Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

PEEKER

Info-Karte

_____

_____

_____

_____

_____

_____

_____

Ich interessiere mich für Beiträge über

☐ Apple ☐ Mac ☐ Atari ☐ _____

Peeker-Börse

Vorname, Name

Firma

Straße

Wohnort

PLZ/Ort

Bitte veröffentlichen Sie den umstehenden Text von _____ Zeilen à _____ DM in der nächsterreichbaren Ausgabe vom **Peeker**

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze

Datum Unterschrift

Produkt-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

Anschrift der Firma angeben, bei der Sie bestellen bzw. von der Sie Informationen wünschen

Umfrage-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

POSTKARTE

Peeker-Börse
Anzeigen-Service

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

POSTKARTE

Inserent

Straße

PLZ/Ort

POSTKARTE

Peeker
Redaktion

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

Produkt-Karte

Wünschen Sie weitere Informationen zu einer der im Heft erschienenen Anzeigen?

Nichts einfacher als das. Produkt-Karte ausfüllen, frankieren und an den Inserenten (nicht an die Peeker-Redaktio senden.

Vorher aber nicht vergessen:
Kreuzen Sie an, welchen Informations-wunsch Sie haben.

Damit erleichtern Sie dem Hersteller eine gezielte Beantwortung Ihrer Anfrag

Zum Schluß tragen Sie auf der Rückseite die genaue Anschrift des Inserenten und Ihren Absender ein.

PEEKER

Seikosha-Drucker im Kurztest

von Andreas Schweitzer

Der Seikosha SP 1000 AP ist ein Imagewriter-kompatibler Matrixdrucker, der durch seinen sehr günstigen Preis von ca. DM 800,– auffällt.

Das Handbuch

Der SP 1000 AP wird mit einem englischen und einem deutschen Handbuch ausgeliefert. Die teilweise schlechte Übersetzung des deutschen Handbuchs erschwert dem Anfänger das Lesen. Das 84seitige Manual enthält alle wichtigen Informationen über den Anschluß des Druckers und eine gute Beschreibung der Steuercodes. Zu jeder Schriftart gibt es Beispielausdrucke.

Inbetriebnahme

Der Drucker ist schnell einsatzbereit, wenn man das richtige Druckerkabel zur Verfügung hat. Ein solches Kabel wird leider nicht mitgeliefert, Imagewriter- oder Scribe-Drucker-Kabel passen aber ohne Änderung. Der Farbbandwechsel erfolgt in Sekundenschnelle und verschont den Benutzer vor schmutzigen Fingern. Die DIP-Schalter befinden sich leicht zugänglich an der Rückseite des Gerätes.

Handhabung

Bei Verwendung von Endlospapier erfolgt die Papierzufuhr von hinten. Dabei sind keine störenden Kabel im Weg. Der Seikosha besitzt automatischen Einzelblatteinzug; für Einzelblätter wird ein Papierhalter mitgeliefert. Bei Papierende wird ein Piepston ausgegeben und der Drucker deselektiert. Diese Option kann auch ausgeschaltet werden.

Schriftarten

Der Drucker verfügt über 11 Schriftarten und 7 nationale Zeichensätze. NLQ und doppelt breite Buchstaben können ebenfalls ausgedruckt werden (s. **Abb.**).

Kompatibilität

Alle Programme, die für den Imagewriter geschrieben sind, laufen ohne Änderung auf dem Seikosha SP 1000 AP (Superdump, Print Shop, Triple Dump usw.).
Mit dem Programm „Font.Loader" kann man ohne Probleme eigene Zeichensätze in den Drucker laden. Hier zeigt sich, daß der Drukker 100% kompatibel ist.

Druckgeschwindigkeit

In Pica-Standardschrift benötigt der Seikosha für 30 Ausdrucke von 79 verschiedenen Zeichen und dem Returnzeichen 43 Sekunden (bidirektionaler Ausdruck). Im NLQ-Modus dauert der entsprechende Ausdruck 210 Sekunden.

Fazit

Wer einen sehr günstigen Imagewriter-kompatiblen Drucker erwerben möchte, ist mit dem SP 1000 AP gut beraten. Der Drucker ist recht sauber verarbeitet. Natürlich müssen einige Abstriche in punkto Materialqualität gemacht werden; anders kann wohl der geringe Preis nicht zustandekommen.

```
DAS IST EXTENDED CHARACTER
DAS IST ELITE
DAS IST PICA PROPORTIONAL

DAS IST SEMICONDENSED

DAS IST ULTRACONDENSED

DAS IST FETTDRUCK
DAS IST DOUBLE WIDTH PRINTING

Und jetzt das ganze in NQL
DAS IST EXTENDED CHARACTER
DAS IST ELITE
DAS IST PICA PROPORTIONAL
```


Public Domain Software
Freiprogramme für Apple Liste sofort bestellen .. 10,-

Software Preissenkung
Wir stellen die Preise auf den Kopf

Merlin Pro Macro Assembler ..... ~~448,-~~ **199,-**
Merlin ..... ~~250,-~~ **150,-**
Merlin Combo **250,-**
Mousewrite .. 498,- **349,-** – der Apple *IIe*® wird zum Mac®

Chart'n Graph Toolbox 110,- **Database Toolbox** **110,-**
Video Toolbox ..... **110,-** **Wizard's Toolbox** **110,-**
Munch A Bug ..... **130,-** **Printographer** **130,-**

ZUSATZ-KARTEN:
V-24-Schnittstelle ..... **199,-** **Z-80-Karte** **98,-**
80-Zeichen-Karte m.Softswitch **236,-** **16 K-Language-Karte** **98,-**
Joy Stick ..... **49,-** **Accelerator** 3,6 MHz **950,-**
68000 Intemex ..... **1600,-** **PAL Karte** **110,-**
RGB Karte ..... **239,-** **IEEE 488** **312,-**
Koppler dataphon m. FTZ ..... **325,-** **Z 80 B Karte** mit Software **919,-**
Centronics-Karte von Epson für Graphik .... **210,-** für Text **145,-**
Centronics-Schnittstelle für 2 Drucker gleichzeitig **129,-**

Super-Eprommer
belegt keinen Slot, incl. Software für 2716-27128 ..... **239,-**

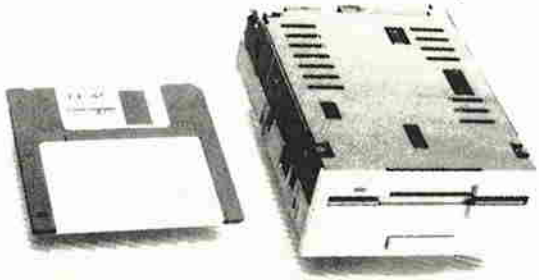
Floppy-Controller
FDC 4 für alle Laufwerke ..... **169,-** Bausatz wie links **159,-**
Leerplatine wie oben incl. Prom u. Eprom **98,-**

Erphi-Controller **298,-**
Disketten 1D, 48 tpi ..... 10 St. **29,-** **Disketten** 2D, 48 tpi ..... 10 St. **36,-**

Frei programmierbare Keyboards
Wir bieten Ihnen die **Preh-Qualität** auch für Apple. AK 87 spez. mit Gehäuse, Anschlußkabel, Zehner-Tastenfeld, dt. Zeichensatz, Sondertasten für Ctrl-Codes und Rechenfunktionen **339,-**
Preh Commander Keyboard, frei programmierbar bis zu 10 Ebenen, pro Taste bis zu 250 Zeichen ..... nur **599,-**
Gleiche Tastatur wie oben **für Apple IIe** ..... nur **698,-**

20 MB Harddisk
(Festplattenlaufwerk) incl. Software, Kabel, Gehäuse etc.
Sonderpreis ..... nur **2998,-**
30 MB und 50 MB für Apple auf Anfrage!

TEAC 3½" Laufwerk FD 35 F 498,-
Speicherkapazität 1 MB, (formatiert 640 KB) jetzt für nur .....



TEAC FD 55 AV 1 x 40 Track .... **395,-** **TEAC FD 55 BV** 2 x 40 Track **459,-**
TEAC FD 55 EV 1 x 80 Track .... **445,-** **TEAC FD 55 FV** 2 x 80 Track **398,-**

Panasonic Drucker:
1091 ..... nur **1095,-** **1592** ..... nur **1599,-**
1092 ..... nur **1195,-**

Die Microfloppy mit Zukunft:
Speicherkapazität: 2 x 1 MByte formatiert: 2 x 640 kByte. Anschlußfertig mit PROM-residenter Patchsoftware für CP/M 2.2, Apple DOS 3.3, DiversiDOS 2-C, 4-C (DD MOVER), Apple Pascal 1.1, Pascal 1.2, Pro-DOS 1.0.1, 1.1, 1.1.1 zum Preis von **1498,-**
Low Power Version 1598,-

Gesamt-Preisliste anfordern!
Preise inclusive gesetzlicher Mehrwertsteuer.
Händlerpreisliste bitte schriftlich anfordern!

UEDING electronics
Holtewiese 2
5750 Menden 1
Telex: (051) 933524 geonet g box IFX2: ueding
DFÜ 02373/66877
Tel. 02373/63159

Die CIRTECH-Flipper-Karte

1MByte RAM für den Apple II+ und IIe

getestet von Hans-Martin Eng

Wer auf einem Apple größere Datenmengen auf einmal zu verwalten hat, darf wählen. Entweder er entscheidet sich gleich für ein Festplattenlaufwerk, oder aber er verwendet eine große RAM-Karte als Pseudo-Diskettenlaufwerk. Derjenige, der sich für letztere Möglichkeit entscheidet, sollte die ca. DM 1150,-- teure Flipper-Karte bei der Auswahl einer geeigneten RAM-Karte nicht unberücksichtigt lassen.
Die Flipper-Karte arbeitet mit den Massenspeichern DiskII, UniDisk, Profile usw. zusammen.
Hergestellt wird sie von der schottischen Firma Cirtech. Der Vertrieb in Deutschland erfolgt über die Firma Semjan Computer Systeme.

1. Lieferumfang

Die Flipper-Karte ist in einem Karton von 26·16·5 cm Größe untergebracht. Darin liegt sie eingebettet zwischen zwei Schaumstofflagen relativ sicher (allerdings nicht vor statischer Elektrizität!). In diesem Karton befinden sich noch eine Diskette mit Installationsprogrammen für die unterschiedlichen Betriebssysteme und die Bedienungsanleitung.

2. Hardware

Die RAM-Karte ist mit 23 cm ziemlich lang ausgefallen. Trotz der hohen Packungsdichte sind die einzelnen Bauteile sichtlich von Hand eingelötet. Da wären im einzelnen 32 RAM-Chips mit je 256 KBit Speicherkapazität, dann 14 Steuerbausteine der 74-LS-TTL-Klasse (die Kennzeichnungen sind zum Schutz vor Raubkopien herausgekratzt*). Betriebssoftware befindet sich auf einem 2732A-EPROM. Schließlich wären da noch diverse Kondensatoren, ein SIL- und ein normaler Widerstand und zwei Abschirmbleche zu nennen. Das ist alles.
In meiner Testkarte waren fast alle ICs noch direkt eingelötet. Weil gerade RAM-Chips doch recht fehleranfällig sind, erleichtert es den Service, sie in Sockeln unterzubringen. Ab sofort wird die Karte deshalb mit gesockelten RAM-Chips ausgeliefert.
Auf das Produkt werden volle 12 Monate Garantie gewährt.

3. Installation

Die Flipper-Karte kann prinzipiell in jedem Slot betrieben werden (außer Slot 0 beim Apple II+ und Slot 3 beim Apple IIe mit erweiterter 80-Zeichenkarte). Will man sie jedoch unter Apple-Pascal 1.3 benutzen, empfiehlt der Hersteller Slot 4, 5 oder 6. Besitzt man einen „Apple mit 65C02-CPU" (Originalzitat), gemeint ist wohl ein enhanced Apple IIe, wird Slot 7 empfohlen, weil man dann von der Flipper-Karte aus booten kann.

4. Ansteuerung

Wie schon kurz erwähnt, ist die Flipper-Karte mit einem EPROM ausgestattet. Auf diesem EPROM befindet sich Software für ProDOS 1.1.1, Apple-Pascal 1.3, Cirtech CP/M Plus und Standard-DOS-3.3. Eine neue Ausgabe des Karten-ROMs soll dann auch Patchsoftware für Appleworks 1.3 enthalten. Schließlich ist noch eine Selbsttestroutine eingebaut.

Die einzelnen Bytes der Karte werden über ein 3-Byte-Register im Softswitchbereich ($C080+$n0 bis $C082+$n0) adressiert. Ein viertes Register enthält dann das Datenbyte ($C083+$n0). Diese Methode erspart das bei größeren Speichern reichlich umständliche Bankswitching. Zusätzlich sind noch Block-Schreib-/Leseroutinen für Pascal und ProDOS im EPROM untergebracht. Diese sind voll zu den Originalroutinen des jeweiligen Betriebssystems kompatibel.

5. Betriebssysteme

5.1. DOS 3.3

Man bootet DOS 3.3 (oder Diversi-DOS, das funktioniert auch) und gibt „IN#n" (n ist die Slotnummer) ein. Kurz darauf stehen zwei Pseudolaufwerke mit je 400K Speicherkapazität zur Verfügung. Die restlichen 200K können nicht genutzt werden. *Leider* funktioniert das nicht bei „gemovten" DOS-Versionen (egal, ob DOS-Mover von Bongers, Grumser oder DSR). Will man seine Flipper-Karte unter „gemovtem" DOS benutzen, muß man sich seinen eigenen RAM-Disk-Treiber schreiben.

Auf der mitgelieferten Diskette befindet sich ein Binärfile namens „SPEEDOS", der das DOS 3.3 in ein schnelleres DOS verwandelt. Offen gestanden reicht mir die Geschwindigkeit von Diversi-DOS vollkommen aus. Ich bezweifle, ob SPEEDOS noch wesentlich schneller sein kann. Nützlich ist SPEEDOS aber für diejenigen, die nur DOS 3.3 und kein Diversi-DOS besitzen.

5.2. ProDOS 1.1.1

Man muß zumindest über ProDOS 1.1.1 verfügen, um die Flipper-Karte benutzen zu können. Dafür muß man sie aber nicht erst extra aktivieren, ProDOS erledigt das automatisch beim Bootprozeß. Dann braucht man nur noch Daten auf die Karte zu transferieren und es kann losgehen.

* Dieses Verfahren ist zollrechtlich dubios. Damit Sie die Karte später exportieren können, brauchen Sie vom Hersteller eine amtlich beglaubigte Stückliste.

Wenn man zusätzlich die Karte mit dem ProDOS-Filer „formatiert" und die Dateien PRODOS, BASIC.SYSTEM und STARTUP darauf transferiert, kann man mittels PR#n ProDOS von der RAM-Karte booten. Das geht dann doch etwas schneller als von einem normalen Diskettenlaufwerk aus. Ist man Besitzer eines enhanced Apple IIe und hat die Flipper-Karte in Slot 7 installiert, wird auch mit Control-Offener-Apfel-RESET von der RAM-Karte aus gebootet.

5.3. CP/M

Für das Cirtech CP/M Plus und andere CP/M-Plus-Systeme befindet sich Betriebssoftware im EPROM der Flipper-Karte. Beim Booten dieses Betriebssystems wird die Flipper-Karte automatisch aktiviert. Für die Softcard-CP/M-Versionen CP/M 2.20B und 2.23 befindet sich auf der mitgelieferten Diskette der File „FLIP.COM", der die Flipper-Karte als Pseudolaufwerk F: bzw. D: installiert.

Für beide CP/M-Versionen gilt, daß beim Warmstart (Control-C) nunmehr das Betriebssystem von der RAM-Karte eingelesen wird; somit geht weniger Zeit verloren. Besitzer eines anderen CP/M (wie ich zum Beispiel) müssen sich einen eigenen RAM-Disk-Treiber schreiben. (CP/M 2.20B, 2.23 und 3.0 sollten jedoch für den normalen Anwender ausreichen.)

5.4. Apple-Pascal

Apple-Pascal 1.3 erkennt die Flipper-Karte automatisch als ein Volume an. Dazu muß sich die Karte allerdings in Slot 4, 5 oder 6 befinden.

Für die Pascal-Versionen 1.1 und 1.2 befinden sich ATTACH-Files auf der mitgelieferten Programmdiskette. Diese müssen auf die Boot-Diskette übertragen werden, aktivieren beim Booten dann allerdings sofort die Flipper-Karte.

5.5. Appleworks 1.3

Auf der Rückseite der mitgelieferten Diskette befindet sich ein Patchfile für Appleworks 1.3. Mit diesem können Sie den Arbeitsspeicher unter Appleworks auf bis zu 1012K erweitern. In Verbindung mit der Karte können z.B. Datenbanken mit über 6000 Datensätzen und Textdateien mit über 7000 Zeilen Umfang erstellt werden. Eine angekündigte neue Version des Flipper-EPROMs soll dann die Modifikation selbst vornehmen.

6. Der Flipper Program Manager

Dieses Programm macht die Flipper-Karte erst so richtig interessant. Mit diesem kann man die RAM-Karte in verschiedene Arbeitsbereiche für unterschiedliche Betriebssysteme einteilen und dann blitzschnell zwischen den verschiedenen Betriebssystemen hin- und herschalten. Außerdem bietet der Flipper Program Manager die Möglichkeit, einen kompletten Arbeitsbereich zur Datensicherung auf Diskette abzuspeichern und wieder zurückzuladen.

7. Die Bedienungsanleitung

Die 24seitige Bedienungsanleitung gibt dem fortgeschrittenen Apple-Anwender erschöpfend Auskunft. Für den Anfänger scheint mir die darin gebotene Kost jedoch noch etwas schwer verdaulich; es wäre nützlich, noch mehr Informationen zu erhalten. Ab sofort wird zur Flipper-Karte auf unsere Anregung eine ausführlichere, 30 Seiten starke Anleitung mitgeliefert.

8. Fazit

Soweit man das durch einen kurzen Test feststellen kann, erscheint mir die Flipper-Karte als ein ausgereiftes Produkt. Besonders hervorzuheben ist, daß sich schon sehr viel Software auf der Karte befindet, so daß ein langer Installationsvorgang bei jedem Booten in der Regel überflüssig ist.

Semjan presents...

• CP/M Plus System für Apple //e,c

- CIRTECH CP/M Plus Modul belegt keinen Slot im Apple //e,c.
- Komplettes Betriebssystem CP/M 3.0 von Digital Research.
- Z80H mit 8MHz, Einsatz von 128K RAM, Drucker-Spooler mit 12K RAM.
- Integration der DiskII und der UniDisk Laufwerke, //e auch Profile.
- Kompatibel zu CP/M 2.20 und 2.23. Apple //c mit Maus-Funktion.
- Mit 20 Hilfsprogrammen, 6 neue CP/M 3.0 System Hilfsprogramme.
- Einsatz von: WORDSTAR, dBASE, MBASIC, TURBO PASCAL 3.0 etc.

K010 //c CP/M Plus System	**DM 798,00**
K011 //c WORDSTAR/MAILMERGE und K010	**DM 1193,00**
K017 //c Turbo Pascal 3.0 und K010	**DM 999,00**
K012 //e CP/M Plus System	**DM 544,00**
K013 //e WORDSTAR/MAILMERGE und K012	**DM 930,00**
K016 //e Turbo Pascal 3.0 und K012	**DM 747,00**
K018 // CP/M 3.0 Programmier-System	**DM 403,00**
K019 //c CP/M Modul V. 2.20 u. 2.23 o. Betr. Sys.	**DM 390,00**

• //+,e CIRTECH 1 MB RAM Karte für AppleWorks 1.2 und 1.3

- CIRTECH FLIPPER Karte wird komplett mit 1 MB RAM geliefert.
- Super schneller Datenzugriff, 50K/sec.. Max. 6 MB RAM pro Apple //.
- 100 % Kompatibel mit PRODOS (Appleworks), DOS 3.3, PASCAL 1.1,
- PASCAL 1.2 und 1.3, CP/M 2.20B, 2.23 und CP/M Plus 3.0.
- Bis zu 1012K RAM Arbeitsspeicher, z. B. mit Appleworks 1.3, etc.
- **AppleWorks 1.3 Update lieferbar!**
- AppleWorks 1.3 Textdatei: 7250 Zeilen. Datenbank 6350 Datensätze.
- AppleWorks 1.3 Rechenblatt: 30 000 Zellen. Einsatz der UniDisk.
- Flipper Programm Manager zum automatischen Verwalten der Karte.
- Laden und Abspeichern des jeweiligen Arbeitsbereiches möglich.
- Einsatz von bis zu 4 Betriebssystemen zur gleichen Zeit!!
- Einsatz von DISKII, UniDisk und anderen ProDos-Block Speichern.
- Kein „patchen" notwendig, Einsatz in jedem Slot möglich.

K070 //+,e Flipper Karte mit 1 MB RAM DM 1148,00

M. Semjan Computer Systeme

Postfach 90 01 64, 6000 Frankfurt/Main 90, Tel. 069-70 18 53
Telex 051 933 521 dmbox g. Ref: Box: DM3:SEMCOM, Mailbox-Adresse: DM3 SEMCOM

Semjan presents...

• Champion Karte für Apple //+,e

- CIRTECH Parallele Text- u. Graphik-Druckerkarte komplett mit Kabel.
- Option 64K RAM Zwischenspeicher, 40/80 Zeichen Dump zu jeder Zeit.
- Umfangreiche Kontroll-Steuerbefehle z.B. Zeichensatzwahl, etc.
- Einsatz von DOS 3.3, PRODOS (Appleworks), UCSD PASCAL, CP/M.
- Champion-Karte voll Graphik fähig, Apple //e Graphik.
- Serieller Ausbau möglich, 75 bis 9600 Baud Ein- bzw. Ausgabe.
- Mischbetrieb, parallel und seriell, zu jeder Zeit möglich.
- Drucker: Epson, Centronics, Brother, Itoh, Imagewriter I/II, etc.

K030 //+,e Champion Interface	**DM 222,00**
K031 //+,e Champion für Imagewriter	**DM 299,00**
K033 //+,e Champion Interface 64K RAM	**DM 476,00**

• Uni-Mate UniDisk Software

- Einsatz der UniDisk Laufwerke ab sofort mit DOS 3.3, CP/M 2.20
- CP/M 2.23, PASCAL 1.1 und PASCAL 1.2. Einmalige Installation.
- Einsatz von UniDisk und DISKII Laufwerken möglich.
- DOS 3.3 bootbar, Speicherkapazität 800K, INIT möglich.
- CP/M 2.2 und 2.23 bietet 768K Speicherk.. Formatierung möglich.
- PASCAL bietet 768K Speicherkapazität. Formatierung möglich.

K045 // CIRTECH Uni-Mate Software DM 127,00

• CIRTECH EPROM SYSTEM

- Super Eprom Programmier-System. System komplett mit aller Software.
- Einsatz als Text/Grafik Interface und Eprom-Programmier-System.
- Einsatz als zweifache I/0 Karte möglich.
- Einsatz von EPROM: 2716, 2732 2732A, 2764, 27128 und 27256.

K080 //+,e Eprom Programmier-System DM 408,00

Auf alle CIRTECH-Produkte volle 12 Monate Garantie.
Neu: Katalog 2/86 anfordern. Händleranfragen willkommen!
Fragen Sie Ihren Händler nach CIRTECH-Produkten!

M. Semjan Computer Systeme

Postfach 90 01 64, 6000 Frankfurt/Main 90, Tel. 069-70 18 53
Telex 051 933 521 dmbox g. Ref: Box: DM3:SEMCOM, Mailbox-Adresse: DM3 SEMCOM

Kyan-Pascal Programming Toolkit

Advanced Graphics

getestet von Matthias Meyer

1. Einsatzgebiet

Advanced Graphics ist ein komplexes Werkzeug für den fortgeschrittenen Pascal- oder Assemblerprogrammierer. Es vereinfacht die Erstellung von anspruchsvoller Grafik-Software. Man hat damit die Möglichkeit, Hires- und Double-Hires-Grafiken herzustellen, die mehrfache Fenster beliebiger Größe und Anzahl enthalten. Das Mischen von Text und Grafik mit verschiedenen Zeichensätzen ist möglich (es wird jedoch nur ein Zeichensatz mitgeliefert). Für die Erstellung von dreidimensionalen Abbildungen wurden leistungsfähige Befehle implementiert. Advanced Graphics läuft nur auf dem Apple IIc oder dem Apple IIe mit der erweiterten 80-Zeichenkarte (64K-Karte).

2. Leistungsumfang

Das Advanced Graphics Toolkit enthält die beiden Module „Graphics Primitives" und „Advanced Graphics".

2.1. Graphics Primitives

Graphics Primitives ist ein von Apple, Inc. entwickeltes, 8K langes Runtime-Modul, welches in den Adreßbereich $7000 bis $8FFF geladen wird. In diesem Modul werden die zu einer Grafik gehörenden Zusatzinformationen in einem sog. *GraphPort* verwaltet: Ein GraphPort ist ein Informationsblock, der die Attributmerkmale der Grafikumgebung enthält. Zur Anpassung dieses Moduls an den Pascal-Compiler hat Kyan, Inc. die folgenden Assemblerprozeduren und -funktionen entwickelt:

2.1.1. Befehle zur Initialisierung

InitGraph: initialisiert die GraphPort-Werte;
SetSwitches: setzt bestimmte Softswitches im Apple zur Einstellung einer bestimmten Grafik-/Text-Konfiguration.

2.1.2. GraphPort-Befehle

InitPort: setzt die angegebenen GraphPort-Attributmerkmale auf Default-Werte;
SetPort: definiert einen angegebenen GraphPort als den momentan zu bearbeitenden GraphPort;
GetPort: ermittelt die Adresse des aktuellen GraphPorts;
SetPortBits: ersetzt das aktuelle GraphPort-Informationsverzeichnis mit den angegebenen Informationen;
SetPenMode: legt die logische Verknüpfungsart beim Zeichnen fest (pencopy, penOR, penXOR, penBIC, notpencopy, notpenOR, notpenXOR oder notpenBIC);
SetPattern: setzt ein 8x8-Pixel-Flächenmuster für nachfolgende Mal- und Zeichenoperationen;
SetColorMasks: setzt die Schablonen, welche die Farbkennzeichenbits beeinflussen – diese Werte werden von nachfolgenden Mal- und Zeichenoperationen verwendet;
SetPenSize: ändert die horizontale und vertikale Breite des imaginären Zeichenstifts;
SetFont: legt den aktuellen Zeichensatz fest;
SetTextBG: legt die aktuelle Farbe des Texthintergrundes fest.

2.1.3. Befehle zum Zeichnen

Move: verschiebt den imaginären Zeichenstift um die angegebene Anzahl Pixeln in horizontaler und/oder vertikaler Richtung;
MoveTo: setzt den imaginären Zeichenstift auf die angegebene XY-Position;
Line: zeichnet eine Linie von der momentanen Position (des imaginären Zeichenstifts) bis zu dem Punkt, der durch die angegebene horizontale und vertikale Verschiebung berechnet wird;
LineTo: zeichnet eine Linie von der momentanen Position bis zur angegebenen XY-Position;
PaintRect: füllt ein Rechteck, von dem die Koordinaten rechts oben und links unten angegeben werden, mit dem aktuellen Flächenmuster aus;
FrameRect: zeichnet den Rahmen um ein Rechteck, verwendet dabei das aktuelle Flächenmuster und die aktuelle Größe des imaginären Zeichenstifts;
InRect: stellt fest, ob die aktuelle Position (des imaginären Zeichenstifts) innerhalb des angegebenen Rechtecks liegt;
PaintBits: zeichnet einen Ausschnitt einer Grafik, die durch das dazugehörige GraphPort-Informationsverzeichnis verwaltetet wird;
PaintPoly: füllt ein Vieleck mit dem aktuellen Flächenmuster aus;
FramePoly: zeichnet den Rahmen um ein Vieleck;
InPoly: stellt fest, ob die aktuelle Position innerhalb des angegebenen Vielecks liegt.

2.1.4. Text-Befehle

TextWidth: berechnet die Breite eines Textes in Pixeln;
DrawText: zeichnet einen Text beginnend mit der aktuellen Zeichenposition.
Dieses Modul unterstützt alle drei Darstellungsmöglichkeiten der Grafik des Apple IIe/c: Hires I (Adressen $2000 bis $3FFF), Hires II ($4000 bis $5FFF) und Double-Hires ($2000 bis $3FFF sind doppelt belegt). Für Assemblerprogrammierer wird vom Runtime-Modul ein sog. Machine Language Interface zur Verfügung gestellt, das dem ProDOS-MLI nachempfunden wurde.

Bewertung: Graphic Primitives ist ein exzellentes Grafikmodul. Die mitgelieferte Proportionalschrift kann sich genauso sehen lassen wie die gut durchdachten Einzelbefehle. Das in Maschinensprache geschriebene Demonstrationsprogramm ist etwas fragmentartig ausgefallen, doch überzeugen die dargestellten Effekte allemal.

2.2. Advanced Graphics

Bei den Advanced Graphics handelt es sich um Befehle, die von Kyan, Inc. zur Erzeugung dreidimensionaler Grafiken entwickelt wurden. Sie unterstützen einfarbige und mehrfarbige Grafikdarstellungen und können dazu verwendet werden, ein Objekt aus einem beliebigen Winkel oder aus einer beliebigen Entfernung zu betrachten. Es ist auch möglich, das Objekt im Raum zu drehen.

Diese Routinen basieren auf dem *CORE-System für Computergrafik* (das CORE-System wurde vom „Graphics Standard Planing Committee of the Association for Computing Machinery" entwickelt, was so viel heißt wie „Komitee der Vereinigung für Datenverarbeitungsgeräte zur Planung eines Grafikstandards"). Das Konzept des CORE-Systems besteht darin, ein Grafiksystem zu entwerfen, das geräteunabhängig ist und auf jedem Computer läuft, der diesen Standard unterstützt.

Die in den Advanced Graphics verwendeten Algorithmen werden in dem Buch „Computer Graphics, A Programming Approach" von Steven Harrington beschrieben (1983, McGraw-Hill Book Company, ISBN 0-07-026751-0). In diesem Buch wird das CORE-System allgemeiner und wesentlich detaillierter behandelt als im Advanced Graphics Manual.

Für Pascalprogrammierer wurden folgende Prozeduren zur Erstellung dreidimensionaler Grafiken implementiert:

Display: mit diesem Befehl werden dreidimensionale Zeichnungen erstellt.
Initialize: setzt die Grundeinstellung für Parallelprojektionen und ruft dann den Befehl MakeViewPlaneXFM auf.
MakeViewPlaneXFM: berechnet die Transformationsmatrix neu und erledigt andere interne Funktionen, die zur späteren Änderung von Prozedurparametern benötigt werden.
SetProjection: wählt zwischen paralleler und perspektivischer Projektion.
SetViewDistance: wählt die „Entfernung" zwischen Objekt und Bildschirm.
SetViewPlaneNormal: setzt die Grundeinstellung für die Betrachtungsweise.
SetViewReferencePoint: stellt die Mitte einer Projektion fest.
SetViewUp: wählt die Ausrichtung des Sichtfensters.
SetWindow: verändert den REAL-Zahlenbereich, der die Grafik innerhalb des Sichfensters erscheinen läßt.

Zur dreidimensionalen Grafikdarstellung wird ausschließlich Double-Hires verwendet. Für einfarbige (monochrome) und mehrfarbige Darstellung gibt es zwei verschiedene Include-Files, so daß die Darstellung je nach verwendetem Monitor stets die maximale Auflösung erreicht.

Bewertung: Wer dreidimensionale Darstellung benötigt, erhält mit den Advanced Graphics ein wertvolles Hilfsmittel. Zur Programmierung von eigenen Anwendungen wird jedoch dringend die Lektüre weiterer Literatur zu diesem Thema empfohlen, da das Kyan-Manual die dreidimensionale Darstellung nur auf 33 Seiten behandelt.

3. Dokumentation

3.1. Äußere Erscheinungsform

Toolkit III von Kyan, Inc. besteht aus einer doppelseitigen Diskette und einer Loseblattsammlung von 61 Blättern zum Einheften in einen Ordner, der das gleiche Format hat wie der des Kyan Pascal User's Manual. Ein von Kyan, Inc. für diesen Zweck vertriebener Ordner kostet US$ 9.95 incl. Porto. Man kann sich aber statt dessen auch einen gewöhnlichen DIN-A5-Ordner kaufen und alle Blätter nochmals lochen.

3.2. Inhalt

Das Advanced Graphics User's Manual beschreibt in einer Einführung die Grundlagen fortschrittlicher Grafikprogrammierung. Begriffe wie „Bitmap, Pixel, Clip Rectangel, Graphics Space, Viewport, View Location und Pen" werden hier anhand von Schaubildern verdeutlicht. Das zweite Kapitel zeigt, wie man das Graphics-Primitives-Modul in Pascal- und Assemblerprogrammen einsetzen kann. Nachfolgend werden die Initialisierungs-, GraphPort-, Zeichen-, Text- und Utility-Befehle beschrieben. Danach folgt ein Verzeichnis der in Pascal verwendeten Datenstrukturen und eine zusammenfassende Liste aller Graphics-Primitives-Befehle. Schließlich folgen das Kapitel über die Verwendung der Advanced Graphics, also der dreidimensionalen Grafikprogramme, und die zugehörige Befehlsliste. Im Anhang werden noch die Farbgrafik und die Diskettenorganisation behandelt.

Bewertung: Die Dokumentation ist sehr ausführlich und gut gegliedert. Die Ausführungen über die dreidimensionale Grafikdarstellung dürften jedoch noch umfangreicher sein.

```
Ein-Blick

Name:                Advanced Graphics Toolkit
Konfiguration:       Apple IIc oder 128K-IIe
                     Kyan Pascal ab Version 2.Ø
Einsatzgebiet:       Grafikprogrammierung in
                     Pascal und Assembler
Gesamtwertung:       *****
Dokumentation:       ****  (118 Seiten)
Zweckdienlichkeit:   *****
Demo-Software:       ****
Verpackung:          ****
Preis/Leistung:      *****
Preis:               US$ 49,95
Hersteller:          Kyan Software, Inc.,
                     San Francisco, U.S.A.

(Die beste Bewertung entspricht 5 Sternen.)
```


Programming Toolkits

für Kyan-Pascal 2.0

Toolkit I: System Utilities: Club-Preis DM 118,-; Normalpreis DM 148,- (lieferbar)

Toolkit II: Mouse Text: Club-Preis DM 118,-, Normalpreis DM 148,- lieferbar

Toolkit III: Advanced Graphics: Club-Preis DM 118,-, Normalpreis DM 148,- (lieferbar)

Toolkit IV: Turtle Graphics: Club-Preis DM 68,-, Normalpreis DM 88,- (lieferbar)

Toolkit V: Mouse Graphic: Club-Preis DM 158,-, Normalpreis DM 198,- (in Vorb.)

Alle Utilities werden als teils beidseitig bespielte Disketten geliefert, die neben den Include-Files (meist Quelltexte) diverse Demos enthalten. Die Anleitungen selbst sind Loseblattlieferungen (z.B. bei den System Utilities 52 Druckseiten), die für den grauen Ordner von Kyan 2.0 bestimmt sind. Zum Club-Preis werden nur Mitglieder des Kyan-Clubs beliefert.

Toolkit I: System Utilities

Diese Utilities decken verschiedene Bereiche ab:
1. ProDOS-Utilities: Delete, Rename, Copy, Set-Pefix, Get-Prefix, Lock, Unlock, Make-Directory, Get-Directory, Remove-Directory, Scan-File, Format, Print-File, Bsave, Bload, Set-Time, Get-Time, Set-Date, Get-Date, Get-Clock, Set-Clock, Find-Clock.
2. Maus und Joystick: Find-Mouse, Init-Mouse, End-Mouse, Home-Mouse, Mouse-Click, Mouse-held, Mouse-moved, Mouse-x, Mouse-y, Set-Mouse-xy, Set-x-Bounds, Set-y-Bounds, Print-Mouse-Char, Button-0, Button-1, Joystick-x, Joystick-y.
3. Bildschirmsteuerung (funktionieren teilweise nur auf IIe/c): Clear-Screen, Clear-Line, Clear-End-of-Line, Clear-End-of-Page, Inverse, Normal, Tab, Scroll-up, Scroll-down, Col-80, On-40, On-80, Screen-Bottom, Screen-Top, Screen-Full, Cursor-x, Cursor-y, Get-Char, Machine-Identification.
4. Zufallszahlen: Random im Bereich min..max, Rnd im Bereich 0..1, Seeding.
5. Zahlenkonvertierungsroutinen: Real – String, String – Real, Integer – String, String – Integer.
6. Sortieren (alphabetisch und numerisch) sowie Mischen (bis zu 5 Files).
7. Line Parsing Routine.

Toolkit II: Mouse Text

Diese Utilities umfassen mehrere Dutzend Befehle für Fenstertechnik usw.
1. Cursor-Befehle: Set-Cursor, Obscure-Curser, Hide-Cursor, Show-Cursor.
2. Interrupts: Check-Events, Get-Event, Post-Event, Set-Key-Event, Flush-Event, Peek-Event.
3. Menü-Befehle: Init-Menu, Set-Menu, Menu-Select, Menu-Key, High-Light-Menu, Disable-Menu, Disable-Item, Check-Item, Set-Mark.
4. Kontrollbefehle: Find-Control, Set-Control-Max, Track-Thumb, Update-Thumb, Activate-Control.
5. Fensterbefehle: Init-Window-Margin, Close-Window, Open-Window, Find-Window, Front-Window, Select-Window, Drag-Window, Grow-Window, Screen-to-Window, Window-to-Screen, Close-all, Window-Char, Window-String, Window-Block, Window-Text, Window-Op.

Toolkit III: Advanced Graphics

Diese Utilities enthalten über 30 Befehle, um Hires- und Double-Hires-Graphiken herzustellen, die mehrfache Fenster beliebiger Größe und Anzahl enthalten:
1. Initialisierungsbefehle: Init-Graph, SetSwitches
2. GraphPort-Befehle: InitPort, SetPort, GetPort, SetPortBits, SerPenMode, SetPattern, SetColorMasks, SetPenSize, SetFont, SetTexBG
3. Zeichenbefehle: Move, MoveTo, Line, LineTo, PaintRect, FrameRect, InPoly
4. Textbefehle: TextWidth, DrawText
5. Befehle zur Erstellung dreidimensionaler Grafik: Display, Initialize, MakeViewPlaneXFM, SetProjection, SetViewDistance, SetViewPlaneNormal, SetViewReferencePoint, SetViewUp, SetWindow.

Toolkit IV: Turtle Graphics

Diese Diskette enthält diverse Hires- und Ton-Routinen.
1. Turtle-Befehle: Init-Turtle, Turtle-x, Turtle-y, Turtle-Angle, Graf-Mode, Text-Mode, Pen-Color, Turn, Turn-to, Move, Move-to, View-Port, Full-Port, Fill-Port, Fill-Area, Save-Hires, Load-Hires.
2. Ton-Befehle: Beep, Note, Clock, Phaser.
3. Balkendiagramme usw.: Bar-Chart, Pie-Chart, Plot-x-y.

Hüthig Software Service
Postfach 10 28 69 Heidelberg

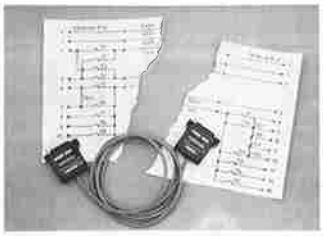
Apple-Produkte

Steckdosen-Störschutzfilter

Moderne leistungsfähige Computer reagieren empfindlich auf sog. Unregelmäßigkeiten des Stromnetzes. Unerklärliche Fehler werden oft durch Hochfrequenzüberlagerungen des Versorgungsnetzes hervorgerufen. Infolge Blitzschlag werden Überspannungen und Stoßspannungen registriert. Die Gerätehersteller treffen Schutzmaßnahmen, soweit sie wirtschaftlich vertretbar sind. Die Kosten für ein mehrstufiges Breitbandfilter würde den Preis aber zu sehr verteuern. Wer auf absolute Betriebssicherheit angewiesen ist, kann sich preiswert mit dem Steckdosen-Störschutzfilter schützen. Auch die auf PCs abgestimmte Version entspricht Industriequalität. Sie besteht aus einem HF-Filter, einem Überspannungsschutz und Stoßspannungsableitern. Der Stückpreis liegt bei ca. DM 50,-.
Bezugsquelle: Metzner-Elektronik, Umkirch

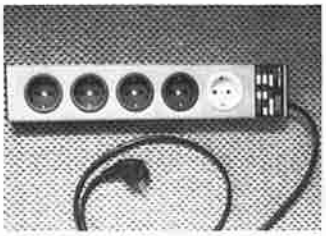
Literaturverwaltung mit Turbo-Lit

„Turbo-Lit" ist ein Literaturverwaltungsprogramm für Mikro-Computer mit den Betriebssystemen CP/M (z.B. Apple mit Z80-Karte) oder PC-DOS (z.B. IBM-PC). Turbo-Lit verwaltet maximal 65536 Einträge (Bücher, Zeitschriften, Aufsätze) und zeichnet sich durch effiziente Suchroutinen, schnelle Verarbeitung und komfortable Menüsteuerung aus. Turbo-Lit bietet u.a. folgende Möglichkeiten: Eingabe und Suche von Stichwörtern, Unterscheidung zwischen Buchtiteln, Zeitschriften und Aufsätzen, Eingabe von 800 Zeichen (CP/M) bzw. 1200 Zeichen (PC-DOS) langen Texten (Abstracts) zu jedem Eintrag, index-sequentielle und assoziative Suche von Schlüsselwörtern, Report- oder Karteikartenformat, automatisches Erstellen von Bibliographien, Kombination verschiedener Schriftarten bei Druckausgabe etc.
Turbo-Lit kostet in der CP/M-Version ca. DM 150,-, in der PC-DOS-Version ca. DM 300,-. Eine Demo-Version kann für DM 20,- bezogen werden.
Bezugsquelle: Bekra-Software, Mülheim-Kärlich



Programmierbares RS232-Kabel

Ein neues Schalterkabel für RS232- bzw. V.24-Schnittstellen hat die Fa. Lindy-Elektronik GmbH entwickelt. Es handelt sich dabei um ein programmierbares, fünfadriges Kabel, das sich für beinahe alle RS232/V.24-Schnittstellen bei Computern, Druckern, Monitoren, Modems usw. eignet. Die Grundidee ist recht einfach: In jedem Stecker sind 10 kleine Schalter eingebaut, die mehr als 1 Mio Schaltvarianten ermöglichen. Lediglich Schutz- und Betriebserde sind fest verbunden (Pin 1/Pin 7). Das 2m lange Kabel trägt beidseitig genormte, 25polige Sub-D-Stecker und kostet nur wenig mehr als ein normales Kabel.
Bezugsquelle: Lindy-Elektronik, Mannheim oder im Fachhandel



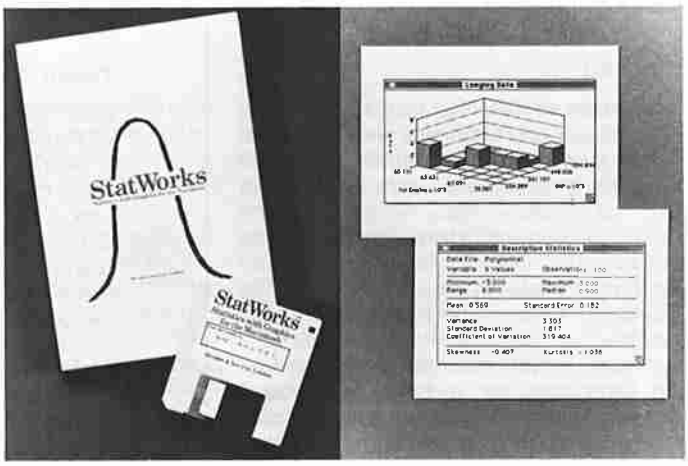
Fünffach-Steckdosenleiste

Wer kennt das nicht? Drucker, Laufwerk, Monitor, die gesamte Peripherie der PCs muß einzeln ein- und ausgeschaltet werden.
Flipp 5 ist eine fünffache Steckdosenleiste zur automatischen Zuschaltung der gesamten Computer-Peripherie. Steckdose Nr.1 ist mit einem Elektronik-Fühler ausgestattet. Sobald durch sie Strom fließt, werden die restlichen angeschlossenen Geräte zeitverzögert zugeschaltet. Der Anwender kann also z.B. nur durch das Einschalten des Bildschirms den Rechner, das Laufwerk, den Drucker und sogar die Tischlampe anschalten. Auf Wunsch wird die Steckdosenleiste mit einem hochwertigen Netzfilter ausgeliefert.
Bezugsquelle: Autotronic GmbH, Bad Kreuznach

Macintosh-Produkte

Silbentrennprogramm MacBlack

Ein Silbentrennprogramm mit den speziellen Trennungen und Trennausnahmen der deutschen Sprache bietet die SCS Schwarz GmbH unter dem Namen „MacBlack" für Macintosh-Anwender an. Setzer mit jahrelanger Erfahrung auf dem Gebiet der automatischen Silbentrennung haben MacBlack für Desktop-Publishing mit PageMaker auf dem Macintosh-Personal-Computer entwickelt. MacBlack verarbeitet ausschließlich MS-Word-Dokumente, in die in einem Arbeitsgang an grammatikalisch zulässigen Stellen Trennfugen eingearbeitet werden. Die so markierten möglichen Trennungen erscheinen automatisch, wenn der Zeilenumbruch eine Trennung erforderlich macht. Für englische Texte oder Texte mit vielen Fremdwörtern kann über Maus-Klick die englische Trennlogik angewählt werden. An einer erweiterten Version, die ein vom Anwender gepflegtes Ausnahmelexikon ermöglicht, wird noch gearbeitet. Preis ca. DM 1120,-.
Bezugsquelle: SCS Schwarz GmbH, Stuttgart



Statistik mit StatWorks

Die Fa. Heyden & Son GmbH bietet mit „StatWorks" ein Statistik- und Grafik-Software-Paket für den Macintosh an. Geeignet ist das Programm v.a. für Wissenschaftler der verschiedensten Fachbereiche, die statistische Analysen benötigen. Das Konzept von StatWorks ermöglicht mactypisches, benutzerfreundliches Arbeiten ähnlich den Programmen MacPaint und MacWrite. Wichtig für alle Mac-Programmierer ist die Möglichkeit, Informationen mit anderen Anwenderprogrammen austauschen zu können. Statistische und grafische Ergebnisse können auf MacPaint übergeben oder in eigene Textdateien übernommen werden. Die Transformationsroutine bietet verschiedene Prozeduren an, um Daten vor der statistischen Analyse zu sortieren und umzuformen oder Daten zu gewichten. Viele Mikrocomputerprogramme sind wegen Rundungsfehlern und unzureichender Genauigkeit (7-14 Stellen) für statistische Aussagen ungeeignet. In StatWorks werden alle Ergebnisse auf 19 signifikante Stellen berechnet. Dies entspricht dem Standard der IEEE und der SANE (Standard Apple Numeric Environment).
Gewisse statistische Vorkenntnisse werden zum Verständnis der vielfältigen Lösungsmöglichkeiten von StatWorks vorausgesetzt. Schließlich entscheidet der Anwender über die richtige Analysenauswahl für sein Datenmaterial.
Zu den statistischen Routinen gehören u.a.: Mittelwert, Varianz, Standardabweichung, Variationskoeffizient, T-Test, Normaltest, Spearman- und Kendall-Korrelationstest, Regressionanalyse, Varianzanalyse.
StatWorks kostet ca. DM 450,-
Bezugsquelle: Heyden & Son GmbH, Rheine

Enhancement-Kit für Lisa

Die Lisa (Macintosh XL) kann zwar im Lisa- und im Mac-Modus betrieben werden, doch mußte man beim Mac-Modus Bildschirmverzerrungen oder ein kleines Bild in Kauf nehmen. Durch das Apple Screenkit wurden diese Nachteile beseitigt, gleichzeitig wurde auch die Kompatibilität zum Macintosh erhöht, dies aber nur unter Verzicht auf die Nutzung der gesamten Lisa-Software. Das Enhancement-Kit bietet auf der Basis des Apple Screenkit weiterhin die freie Auswahl zwischen den Betriebssystemen und den Bildschirmdarstellungen. Es erschließt damit für die meisten Programme die Vorteile des größeren Bildschirms. Damit ist in beiden Betriebsarten eine optimale Bildschirmdarstellung und Ausnutzung gewährleistet.
Bezugsquelle: R.Schlösser, München

Mac-Aufrüstung

Die wesentlichen Vorteile des Mac Plus können auch auf dem Macintosh 512K genutzt werden, nämlich das Hierarchische File System (HFS), optimiertes QuickDraw für schnellere Grafik und RAM-Caching für optimales Arbeiten mit dem Finder und System.
Ermöglicht wird diese Leistungssteigerung durch die neuen ROMs und ein 800K-Laufwerk von Apple. Für Besitzer eines Macintosh mit 128K bietet die Fa. Schlösser mit ihrer Speichererweiterung einen günstigen Einstieg in diese neue Leistungsklasse für unter DM 300,– an. Diese Speichererweiterung erschließt dem Anwender die Vorteile leistungsstarker Programme wie Jazz, Excel etc.
Bezugsquelle: R. Schlösser, München

Telexen mit NetMaster

Die Firma GNT Automatic GmbH bietet mit dem GNT NetMaster 2001 ein System an, mit dem Macintosh und Apple II als Telexgerät benutzt werden können. Der Computer wird dabei mit einem Kommunikationsprogramm (MacTerminal/Access II) an den GNT NetMaster angeschlossen. Der NetMaster ist ständig sende- und empfangsbereit. Er ermöglicht es, Telexnachrichten im Dialog oder vorbereitete Texte zu senden und speichert ankommende Telexnachrichten ab, bis sie vom Computer aus abgerufen werden. Besondere Funktionen des NetMaster: Rundsendungen an mehrere Empfänger mit persönlicher Anrede, Sammeltexte an einen Empfänger, Rufwiederholung entsprechend der Fernmeldeordnung, Statusabfrage, Journal (Protokollmodus). Preis: NetMaster 2001 mit 64K ca. DM 7900,-, mit 128K ca. DM 9000,-.
Bezugsquelle: GNT Automatic GmbH, München

COBOL-Compiler

MacCOBOL von Micro Focus stellt die erste COBOL-Hochleistungs-Programmierumgebung für den Macintosh dar. MacCOBOL umfaßt einen ANSI-74-Compiler, der die Vorteile des Macintosh ausnutzt und dem Anwender erlaubt, mit einem Maus-Klick Programmierhilfen anzuwählen. MacCOBOL wurde für den professionellen COBOL-Programmierer entwickelt und ermöglicht es, auf dem Macintosh ausgefeilte Anwendungsprogramme für die Wirtschaft zu erstellen.
Die MacCOBOL-Programmierhilfen umfassen: EDIT – einen Texteditor, COMPILE – einen Hochleistungs-ANSI-74-Compiler, GENERATOR – erzeugt den Micro-Focus-Zwischencode zum 68000-Objektcode, RUN – ermöglicht die Ausführung zahlreicher unabhängiger Programme und BUILD – erzeugt Module, die Anwendungsprogramme mit dem Run-Time-System verbinden.
Konfiguration: Apple Macintosh oder Lisa 2 (XL), mind. 512K, externer Festplattenspeicher empfohlen.
Das System kostet ca. DM 1480,-.
Bezugsquelle: Micro Focus GmbH, München

Leserbriefe und Ergänzungen

TurtleGraphics-Library-Pakets

Seit Herausgabe des Pakets sind mehrere Leserbriefe eingetroffen, die von einigen Schwierigkeiten im Umgang mit dem Paket berichten, die mir selbst nicht aufgefallen sind.

Stack-Overflow

Die TurtleGraphics hat durch ihre große Anzahl von Prozeduren und Funktionen einen langen Interface-Text. Alle darin erklärten Bezeichner werden bei einem „uses TurtleGraphics" übernommen. Das kostet beim Übersetzen sehr viel Speicherplatz, wodurch der Fehler „Stack overflow" auftreten kann, wenn man nicht gerade die 128K-Version des Pascal 1.2 benutzt. Abhilfe schafft das Einschalten der Swapping-Option des Compilers, also ein „{$S+}" in der ersten Zeile des Programms. Dadurch erhöht sich zwar die Übersetzungszeit, aber es können außer der TurtleGraphics auch noch andere Units benutzt werden, ohne daß der Speicherplatz beim Übersetzen knapp wird.
Schwierigkeiten gab es auch beim Zusammenstellen der Disketten laut Anleitung Seite 2. Wenn beim Booten die Meldung „Must Link first" erscheint, wie das in einem Fall auftrat, so befindet sich vielleicht ein ungelinktes SYSTEM.ATTACH oder SYSTEM.STARTUP auf der Boot-Disk. Sonst kann die Meldung nicht auftreten. Diejenigen, die es nicht schaffen, sich die Arbeitsdisketten herzustellen, sollten zwei Disketten mit ihrem Betriebssystem an den Hüthig-Verlag senden. Auf diese werden dann die Files kopiert.

Softswitches

Das Programm TurtleTest auf Seite 6 des Manuals ist nicht hundertprozentig richtig. Ich vergaß, als letzten Befehl ein „EndTurtle" einzubauen, so daß die Softswitches wieder restauriert werden. Wird das nicht getan, so bleibt der Rechner gemäß Seite 2 im 40-Zeichen-Modus, da nur so die zweite Seite dargestellt werden kann. Beim Zurückkehren aus dem Programm würde dann die 80-Zeichenkarte abgehängt werden. Des weiteren erscheint die Schrift in der Mitte des Bildschirms, da die Position der Schildkröte bei „InitTurtle" auf die Mitte initialisiert wird. Man sollte vorher ein „MoveTo (0, 100)" einbauen. TurtleTest sieht dann folgendermaßen aus:

```
program TurtleTest;

uses TurtleGraphics;

begin
  InitTurtle;
  InitPage2;
  ShowPage (1);
  MoveTo (Ø, 1ØØ);
  WString ('Das kann man zuerst
            nicht sehen');
  readln;
  ShowPage (2);
  readln;
  EndTurtle
end.
```

Außerdem muß erwähnt werden, daß der P-Code-Interpreter von Pascal 1.2 (128K-Version) ständig die Softswitches benutzt, um den P-Code zu holen; das BIOS (Prozedur CONCK) schaltet ebenfalls ständig um. Deshalb ergibt sich bei der 128K-Version des Pascal 1.2 ein flimmerndes Bild. Der Fehler liegt aber nicht in der TurtleGraphics, sondern im Betriebssystem. Deshalb sollte man auf die Benutzung der zweiten Hires-Seite verzichten. Benutzt man aber die 64K-Version von Pascal 1.2 oder Pascal 1.1, so funktioniert alles einwandfrei.

Demo Print

Die Ausgabe des Demo-Programms „Print" erhält man nur bei Benutzung der doppelt hochauflösenden Grafik, wie man aber deutlich aus dem mitgelieferten Programmtext erkennen kann. Besitzt man nur einen Apple II+, so stürzt das Programm nicht ab, aber die Buchstaben erscheinen doppelt so breit, so daß nicht alle Zeilen komplett auf dem Bildschirm dargestellt werden. Der Befehl „InitDouble" wird dort in den Befehl „InitTurtle" umgewandelt. Welches Demo-Programm jeweils die doppelt hochauflösende Grafik benutzt, kann man leicht aus den mitgelieferten Quellen herausfinden („InitDouble" statt „InitTurtle").
Ich hoffe, Ihre Leserbriefe ausreichend beantwortet zu haben.
Dieter Geiß

PT88-Drucker-Antworten

Screen-Dump für Siemens PT88

Für Besitzer eines Apple IIe oder IIc, die einen Siemens-Drucker PT88 benutzen, sind folgende Änderungen des Programms Screen-Dump auf Kyan Pascal, Disk 2, Side 1; /KIX/BIN/SD nötig:
Die Zeile „0917 C950 CMP #$50" muß umgewandelt werden in „0917 C94F CMP #$4F".
Das Programm DUMP80REL aus Peeker, Heft 4/86, S.52 kann auch für den PT88 angepaßt werden. Dazu muß Zeile 50 verändert werden: Statt „50 232,224,80...." muß es heißen „50 232,224,79....".
Getestet wurden diese Änderungen auf meinem Apple IIc mit NZ80-IIc-Karte, zweitem Laufwerk und Drucker Siemens PT88.
U. Metzdorf, Langenhagen-Godshorn

Apple-Anschluß für PT88

Im Peeker, Heft 5/86, S.72 fand ich in der Rubrik Leserbriefe die Anfrage von Herrn Jörg aus Gerlingen bezüglich des Siemens Tintenstrahldruckers PT88. Seit zweieinhalb Jahren benutze ich ausschließlich diesen Drucker in Verbindung mit dem Parallelinterface Grappler+, ursprünglich am Apple II+, jetzt am IIe. Mit seinem Grundbetriebsprogramm kann der PT88 in dieser Konfiguration aber nur die normale Textausgabe unterstützen, die Grafikroutinen im Grappler+ setzen den „Epson-Standard" voraus. Hier bietet die Firma IBS mit der AP4G und einem speziellen PT88-EPROM Unterstützung an. Eine andere Möglichkeit, den PT88 mit der restlichen Applewelt zu verbinden, besteht darin, ein anderes Grundbetriebsprogramm zu implementieren. Von der Firma Siemens kann ein Epson-, ein IBM- oder ein BTX-Modul erworben werden. Dieses Modul besteht jeweils aus zwei EPROMs für die Hauptspeicherplatine und der Speichererweiterung (4K) nebst EPROM für den Zeichengenerator. Ich verwende seit einiger Zeit das Epson-Modul und bin seitdem aller Kompatibilitätsprobleme ledig. The Print Shop z.B. arbeitet absolut korrekt. Ich unterstütze gerne andere PT88-Besitzer.
H. Baars, Wolfenbüttel

Imagewriter

Imagewriter-HTAB

Bei PRINT TAB(5)"A";TAB(10)"B" wird auf meinem Imagewriter (mit IIc) „B" nicht in der 10. Spalte – wie auf dem Bildschirm üblich, sondern in der 15. Spalte (d.h. 5+10 = 15) ausgedruckt. Ist dieses Verhalten beim Imagewriter normal oder ist bei meinem Imagewriter etwas falsch eingestellt? Bei aus den Programmen errechneten Argumenten für TAB ist das besonders lästig.
M. Czudnochowski, Leverkusen
Antwort: Der TAB-Befehl funktioniert zwar am Bildschirm, jedoch nicht beim Imagewriter. Es gibt folgende Lösungsmöglichkeiten:
a) Man verwendet statt TAB (X) den Befehl POKE 36, X-1.
b) Man schaltet zum Ausdrucken das Bildschirmecho ein (mit Ctrl-A L; das N-Flag der Speicherstelle $06B8 des IIc muß gesetzt sein.)
In beiden Fällen funktioniert dann die Tabulierung korrekt. us

Imagewriter-HTAB

Dem Artikel „Imagewriter kurz und bündig" im Peeker, Heft 12/85 möchte ich folgendes zufügen: Auch ich habe den Eindruck, daß sich der Verfasser des Handbuchs zum Imagewriter nicht genug mit diesem Drucker beschäftigt hatte. Deshalb bin ich froh über den Artikel, der manches klärte.
Leider ging Herr Schunk nicht auf den Tab-Befehl ein. Auch er scheint nicht so zu funktionieren, wie es beschrieben ist. Deshalb wäre ich sehr dankbar, wenn Sie mir bei der Anwendung des HTAB-Befehls helfen könnten (UCSD-Pascal oder BASIC).
H. Westram, Monschau

Kyan-Pascal

Kyan-Pascal-Aufbaukurs

Herzlichen Glückwunsch zu Ihrem Entschluß, mit Kyan-Pascal diese Sprache bekannter machen zu wollen. Die Art Ihres Aufbaukurses (Heft 3/86) läßt mich allerdings vermuten, daß Sie in Pascal eine fortschrittliche Abart von BASIC sehen. Über wissenschaftliche „Eindeutigkeit" in einer volkstümlichen (populären, lat. populus = Volk) Zeitschrift läßt sich gewiß streiten. Sie werden aber zugeben, daß „homogen" sich später einfacher liest als „mit Elementen vom gleichen Typ", wenn man sich den bestimmenden (definierenden) Satz einmal gemerkt hat. Dieser klingt tatsächlich seltsam, wenn man ihn aus dem Zusammenhang nimmt.
(Nicht an dem Fremdwort „homogen" nahm ich Anstoß, sondern an der tautologischen Definition. us)

Aufzählung und Menge („enumeration" und „set") haben durchaus ihre Anwendungen, wie die folgenden Pascalzeilen zeigen sollen:

```
TYPE  symbols = (Arraysym,
Beginnsym, Casesym,...Whilesym);
SymSet = SET OF symbols;

VAR sym: symbols;
    StatmBeginSym: SymSet;
IF sym = Repeatsym THEN BEGIN
   GetSym(sym);
   WHILE sym IN StatmBeginSym
   DO BEGIN
      statement(sym);
      GetSym(sym);
   END;
   IF sym:= UntilSym THEN...
```

Obiges Beispiel stammt aus einem Pascal-Compiler, der natürlich in Pascal geschrieben ist. Solche „nicht-infantilen" Programme werden von Elementen des Typs N. Wirth erstellt. Die erwähnten „speziellen Befehle" sind in diesem Fall Mengenmanipulationen, welche in jeder Stufe der Compilation die erlaubten von den nicht erlaubten Symbolen trennen. Ich habe in 10 Jahren Pascal-Programmierung noch kein Überschreiben des Programmspeichers erlebt (in FORTRAN passierte das schon mal). Die Laufzeittests haben immer „Stack overflow" oder „Heap overflow" gemeldet. Bei Turbo Pascal kommt die Laufzeitgeschwindigkeit allerdings auch nicht von ungefähr.
Ich bin gespannt auf die Kyan-Implementation!
Ch. Wehrli, Davos

(Antwort: Zum Stack-Overflow gestatte ich mir noch folgenden Zusatz: In UCSD beispielsweise wird bei einem Stack-Overflow der Exit-Code brutal mitten in den Speicher „reingehauen". Danach „darf" man Reset drücken. Verwendet man eine Accelerator-Karte und 128K-UCSD-Pascal, so muß man sogar den Rechner ausschalten, weil sich UCSD-Pascal nach einem Stack-Overflow nicht einmal bemüht, die 64K-Karte abzustellen. us)

Kyan-Pascal-Strings

```
PROGRAM XYZ;
TYPE String = ARRAY [1..10] of Char;
VAR Word: String
BEGIN
Word := 'abc' ...
```

ergibt „not assignment compatible". Warum?
Paul Batt, Zürich
(Antwort: Die Zuweisung einer Stringkonstanten in einfachen Anführungszeichen muß rechts mit so vielen Leerzeichen aufgefüllt werden, wie der definierte String lang ist, also hier Word := 'abc-------'; „-" = Space. Bei UCSD- und Turbo-Pascal ist dies deshalb nicht erforderlich, weil dort Strings bereits vordefiniert sind. us)

Kyan-PR(1) und 80 Z/Z

Mehrfach wurde von IIe-Besitzern gefragt, warum nach dem Abschalten des Druckers mit PR(0) der 80-Zeichenbildschirm abgeschaltet oder nur noch in jeder zweiten Spalte scrollt. Hierzu vorab ein Experiment:
Schalten Sie den Apple IIe aus und wieder ein, ohne vorher eine Diskette ins Laufwerk gelegt zu haben, und drücken Sie dann Reset. Sie befinden sich nunmehr in demselben I/O-Zusatz wie Kyan-Pascal. Wenn Sie jetzt PR#3 eingeben (80 Z/Z an) und dann PR#1 (Drucker an), dann haben Sie denselben Zustand wie nach Kyan-PR(1). Ob nunmehr PR#0 oder Kyan-PR(0) entweder auf 40 Z/Z umschaltet oder einen „halben" 80-Zeichenbildschirm liefert, hängt davon ab, ob Sie die alten oder die neuen IIe-ROMs benutzen. Die „Quick-and-dirty"-Lösung lautet für Kyan-Pascal bei beiden IIe-Typen: War die 80-Zeichenkarte vor PR(1) bereits eingeschaltet, so wird der Drucker mit PR(3) ausgeschaltet, andernfalls mit PR(0). Nachteil: PR(3) löscht den Bildschirm (ähnlich wie Ctrl-L). Wenn dies unerwünscht ist, so wird es komplizierter: Man rette vor PR(1) durch Peeks die Speicherstellen $0036-$0037. Nach Beendigung des Druckvorgangs poke man die geretteten Werte wieder zurück. PR(0) ist dann überflüssig. us

Diverse Leserfragen

MMU-RAM-Disk beim IIc

Nach Installation einer beliebigen RAM-Disk aus MMU 2.0 kann auf Apple IIc der Drucker nicht mehr angesteuert werden. Der Zustand bleibt so lange erhalten, bis der Rechner aus- und wieder eingeschaltet wird. Neubooten, auch Neubooten von ProDOS, hilft nicht.
P. Förster, Essen

(Antwort: Der IIc benutzt für seinen Druckertreiber auf der 64K-Karte nicht nur den Bereich ab $0800 als Druckerpuffer, sondern zu allem Übel auch noch deren Zero-Page. Die verschiedenen RAM-Disk-Driver für DOS 3.3 auf der MMU-Diskette liegen in der 64K-Zero-Page. Auf diese Weise lassen sich wegen des komplizierten Bank-Switching die schnellsten RAM-Disk-Driver schreiben, die zudem die kompletten 64K ausnutzen. Beim IIc müßte der RAM-Disk-Driver teilweise in den unteren 64K liegen. Dies führt jedoch wieder zu Konflikten mit Anwenderprogrammen. Daher habe ich davon abgesehen, den RAM-Disk-Driver für den IIc umzuschreiben. Unter ProDOS liegen die Dinge anders, denn dort liegt ein Teil des von Apple eingebauten RAM-Disk-Drivers in der Language-Card. Aber auch bei dem ProDOS-RAM-Disk-Driver kann es bei Interrupts Probleme geben, nämlich dann, wenn sich RAM-Disk-Daten im Bereich der Interrupt-Vektoren, also ab $FF00, befinden. Fazit: Auf dem IIc ist eine 64K-RAM-Disk unter keinem Betriebssystem problemlos installierbar, weil die oberen 64K auch anderweitig benutzt werden. us)

Serielle Schnittstellen beim IIc

Ich lese regelmäßig den Peeker, da Sie sehr viele interessante Beiträge für Apple-II-Besitzer veröffentlichen. Zu dem Artikel „Konfiguration der seriellen Schnittstellen beim Apple IIc" von E. Klatt (Peeker, Heft 2/86, S.34) habe ich allerdings noch einige Rückfragen . Die Initiierung der Ports ist verständlich. Wie bekomme ich nun aber Daten über die Kommunikationsschnittstelle als Datei (ASCII) in den IIc bzw. auf Diskette (z.B. Programme aus Nicht-Apple-Computern lesen oder abspeichern)? Baudrate, Datenformat, Parität, Pegel usw. sind aufeinander abgestimmt. Der Programmieraufwand sollte möglichst gering sein (Basicprogramm), und die Datenübernahme sollte auf dem Monitor verfolgbar sein.
M. Jud, Mainz

IIe-INTROM-Dump

Ich besitze einen Apple IIe, der in den ROMs u.a. auch eine Self-test-Routine enthält. Nach dem Handbuch liegt diese im Monitorbereich $C100 bis $C2FF und $C400 bis $C7FF. Ich möchte mir diesen ROM-Bereich als Hex-Dump auf dem Bildschirm auflisten, finde aber dazu keine Angaben im Handbuch.
A. Stirba, Erlangen
Antwort: Das interne CX-ROM ($C100-$CFFF) wird nach dem Aufruf der jeweiligen Routine (von $F800ff. aus) regelmäßig wieder ausgeblendet, weil sonst die Slot-ROMs der Interface-Karten sich selbst nicht mehr lesen könnten. Mit

```
CALL -151
C007:0
1100<C100.CFFFM
C006:0
```

können sich sich das komplette CX-ROM in den Bereich $1100-$1FFF kopieren und dann ansehen oder auf Diskette speichern. Ergänzend sei darauf hingewiesen, daß die (umfänglichen) Selbsttestroutinen nur im alten IIe enthalten sind und nicht in den neuen ROMs, der (aus Platzgründen) quasi nur noch einen Mini-Selbsttest enthält. us

Modembetrieb mit Apple IIc
Die schon reichlich empfangenen und verdienten Lorbeeren für Ihre Zeitschrift möchte ich unterstreichen mit dem Hinweis, daß mir an Ihren Beiträgen besonders auch die kritische Auseinandersetzung mit den Software-Produkten der Firma Apple imponiert. Dies kann man ja letztlich nur, wenn man auch konstruktiv Besseres zu bieten hat – was Sie mit Ihren ProDOS-Utilities ja schon bewiesen haben. Als Anstoß möchte ich Ihnen ein weiteres Beispiel für die merkwürdige Politik der Firma Apple geben, die geradezu nach einer kritischen Auseinandersetzung „schreit". Leider bin ich trotz des guten Artikels im Peeker 2/86 über die seriellen Schnittstellen des Apple IIc mangels Kenntnissen nicht selbst dazu in der Lage. Hier mein Beispiel aus dem Apple-Handbuch zum IIc (Systemdienstprogramme, S.43):
„Wichtig! Aufgrund von Besonderheiten des Apple-IIc-Baudratengenerators erfordern einige Akustikkoppler ein anderes Datenformat als im Handbuch für das Übertragungsnetz angegeben ist. Daher sollten andere Datenformate ausprobiert werden, bis das richtige Datenformat gefunden ist."
Was hat es mit diesen Besonderheiten auf sich? Ich habe z.B. festgestellt, daß im Betrieb mit dem Datex-P-Netz der Post solche „Besonderheiten" wirksam werden und lange gebraucht, bis ich dahinter kam, welche Konfiguration im „Terminal-Modus" des IIc und bei Verwendung des Terminalprogramms benötigt wird. Muß man das wirklich der „Trial-and Error-Methode" überlassen?
Vielleicht könnten Sie sich entschließen, den Bereich der Datenkommunikation doch nicht so „unterbelichtet" wie bisher zu lassen. Sie haben zwar schon mitgeteilt, daß Sie mangels einer entsprechenden Hard- und Software-Ausrüstung in der Redaktion nicht in der Lage sind, sich mit Fragen der Datenkommunikation zu befassen, aber sicher wäre jemand aus der Leserschaft bereit, Ihnen Software und einen Akustikkoppler für Tests zur Verfügung zu stellen.
S. Dach, Blaubeuren

CP/M-Ctrl-K
Ich besitze einen Apple II+-kompatiblen Rechner, der u.a. mit einer 80Z-Karte der Fa. Ehring arbeitet. Nun habe ich folgendes Problem: Unter CP/M erscheint beim Drücken der Taste CTRL-B ein „Ä" und bei CTRL-K ein „Ö" auf dem 80-Zeichen-Bildschirm. Gerade diese Zeichen werden aber von einigen Programmen benötigt, um gewisse Funktionen auszuführen. Anstatt nun dies zu tun, erscheint wie oben erwähnt nur ein „Ä" oder „Ö". An der Tastatur kann es nicht liegen, da diese die richtigen ASCII-Codes an den Rechner liefert (CTRL-B: 02H, CTRL-K: 0BH). Unter Applesoft-BASIC liegt dieses Verhalten nicht vor.
Hußfeldt, Lübeck
(Antwort: Man muß unter MBASIC das Programm CONFIGIO starten und dann unter dem Menüpunkt 2 die Umdefinition der Tasten rückgängig machen. Da man aber jetzt noch Ctrl-K nicht über die Tastatur direkt eingeben kann, muß man statt dessen &H0B eingeben.)

ONERR-Befehl
Gibt es eine Möglichkleit, in Applesoft-Basic einen „ONERR GOTO XY"-Befehl rückgängig zu machen, so daß ich im 1. Teil eines Programms Fehler durch eigene Unterprogramme bearbeiten und später wieder die Interpreter-Fehlermeldungen benutzen kann? Ist es möglich, nur einige bestimmte Fehlersorten selbst abzufangen?
K. Branmann, Kaarst
Antwort: Mit POKE 216,0 kann ONERR insgesamt, aber nicht selektiv in bezug auf bestimmte Fehlerarten, wieder abgestellt werden. Näheres steht im Applesoft-Manual. us.

Errata

Superdump
Das Programm SUPERDUMP aus Peeker, Heft 6/85, S.25 ist auf dem Apple IIc auch mit der Änderung aus Peeker 8/85, S.50 nicht lauffähig. Statt POKE 34499,32 muß es heißen: POKE 34487,32. Aus ‚$86B6: AND #$40' wird dadurch ‚AND #$20' (Druckerstatus).
W. Schreiter, Wien

Reelle Zahlen
Das Programm „Reell.6" aus dem Artikel „Behandlung reeller Zahlen in Applesoft" aus dem Peeker, Heft 3/86, S.12 ist nicht ganz vollständig. Zur Komplettierung der Umwandlung muß für Zahlen -1 < x > 1 und für x <> 0 die Mantisse vom dritten bis zum fünften Schritt als „kleiner/gleich 128" erscheinen. Wenn man die Zeilen 490 und 500 dementsprechend abändert, fließt das Programm in das (korrekte) Endergebnis ein. Im vierten Schritt sollte man das Vorzeichen nochmals beibehalten; dies läßt sich in Zeile 520 ändern.

```
49Ø INVERSE: PRINT V$;: E(7) = 1:
    IF E$ = "-" THEN FOR J = 1
    TO 7: E(J) = ABS (E(J)-1):
    NEXT: IF E(Ø) = Ø THEN FOR
    J = 1 TO 7: E(J) = ABS
    (E(J)-1): IF E(J) <> 1
    THEN NEXT
5ØØ GOSUB 68Ø: NORMAL: PRINT" ";
    GOSUB 69Ø
52Ø PRINT V$;: GOSUB 68Ø:
    FOR J = 1 TO I + 31:
    IF INT((J-1)/8) = (J-1)/8
    THEN PRINT " ";
```

A. v. Lier, München

Quicksort
Nach Aufruf des Programms Quicksort von Harald Grumser im Peeker, Heft 1/86, S.18 meldet sich der Computer mit der Fehlermeldung „OUT OF DATA ERROR", sobald im Basicprogramm versucht wird, irgendeine Variable zu verwenden. Dieser Fehler tritt nur dann *nicht* auf, wenn vorher eine FOR-NEXT-Schleife eröffnet wird. Das liegt daran, daß Speicherstelle $14 (Subflag) zu Beginn des Programms mit $40 (Feldvariable) belegt wird. Diese Speicherstelle muß vor Beendigung des Sortierprogramms wieder mit $00 belegt werden (normale Variablenbehandlung).
Abhilfe wird geschaffen durch Einfügen der folgenden Zeilen, beginnend mit Zeile 102, ins Assemblerlisting:
102 LDA #$00
103 STA SUBFLAG
Bei vorheriger Eröffnung einer FOR-NEXT-Schleife tritt dieser Fehler deshalb nicht auf, weil diese ihrerseits diese Speicherstelle verändert und anschließend mit dem „richtigen" Wert $00 belegt.
H. Maier, Weißenburg

Wordstar-Drive-Wechsel
Im Peeker, Heft 6/86, S.68 beantwortete Jörg Bliesener einen Wordstar-Leserbrief. Darin erstellte er ein kurzes Assemblerprogramm, um Drive B: als sofort aktuelles Laufwerk aufrufen zu können. In dieses Programm hat sich ein Fehler eingeschlichen. In Spalte 4, Zeile 1 lautet die richtige Version „02A4H CDH" anstelle von „02A4H CBH".

MMU 2.0
Memory Managements Utilities
für die Apple IIe 64K-Karte
DOS 3.3 (und ProDOS)
von U. Stiehl
1984, Diskette und Manual, DM 98,–
ISBN 3-7787-1023-1
Insgesamt enthält die neue „MMU 2.0"-Diskette über 25 Programme, die neue Einsatzmöglichkeiten für die Extended 80 Column Card (erweiterte 80-Z-Karte = 64K-Karte für den Apple IIe) erschließen. Ein Teil der Programme laufen auch auf dem Apple II Plus, doch ist „MMU 2.0" primär für 64K-Karte-Besitzer gedacht.
Gerätevoraussetzung: Apple IIe mit 64K-Karte oder IIc
Hüthig Software Service,
Postfach 10 28 69, D-6900 Heidelberg

Ältere Peeker-Hefte
können für DM 6,50 pro Heft zuzüglich Versandspesen angefordert werden. Vergriffene Hefte sind als Photokopien für DM 10,– pro Heft erhältlich. Mindestbestellmenge 2 Hefte.
Dr. A. Hüthig Verlag · Heidelberg

Apple GS und Atari ST im Vergleich

von Ulrich Stiehl

1. Allgemeines

Wie Sie bereits dem Editorial des September-Heftes entnehmen konnten, wird der Peeker in Zukunft neben den Apple-II-Rechnern auch die Computer der Atari-ST-Serie behandeln. Wir bringen deshalb im folgenden keinen isolierten Testbericht zum Apple II GS, sondern einen Vergleich mit dem Atari 1040 ST, damit die Vor- und Nachteile der beiden Computer deutlicher zutage treten.

Der Peeker hat bislang neben den Grundlagenbeiträgen zur Informatik und den Kompaktkursen über Mikroprozessoren und Programmiersprachen, die für einen allgemeinen Leserkreis bestimmt sind, ausschließlich Aufsätze für Apple-II-Besitzer veröffentlicht. Wie Ihnen jedoch bekannt ist, wird der IIe/c von Apple Deutschland seit über zwei Jahren werblich ignoriert. Kaum ein Peeker-Leser wird sich noch an irgendeine der Apple-II-Anzeigen erinnern können, die früher häufig erschienen sind. Auch auf Messen sucht man jetzt meist vergeblich nach einem Apple IIe/c. Es verwundert deshalb nicht, daß die Apple-II-Verkäufe spürbar zurückgegangen und kaum noch neue Fremdprodukte entwickelt worden sind.

Auch der Apple-II-Nachfolger wird, wie uns versichert wurde, keine verkaufspolitische Priorität genießen und deshalb kein forciertes Marketing erfahren, denn in München hat man sich inzwischen auf Spezialmärkte eingerichtet, z.B. auf die Satztechnik mit dem Macintosh. Wir meinen demgegenüber, daß der neue Apple, insbesondere angesichts des hohen Preises, ohne massive Werbung keine nennenswerten Verkaufszahlen erzielen kann. Wir werden deshalb zunächst einmal abwarten, was sich an der Verkaufsfront tut. Käufer des neuen Apple-Computers werden gebeten, uns per Postkarte o.ä. ihr Interesse an entsprechenden Aufsätzen zu bekunden.

2. Kurzüberblick

Apple GS

Der neue und ab Oktober 1986 erhältliche Apple II nennt sich Apple II GS oder kurz Apple GS (G = Graphics, S = Sound). Lange Zeit wurde der Arbeitstitel IIx verwendet, weil man sich noch nicht über das Kürzel einigen konnte. Firmenintern nannte man ihn Cortland, was wohl der Name für eine Apfelsorte ist.
Das Grundgerät besteht aus einer Konsole bzw. Platine in einem Kunststoffgehäuse und einer separaten Tastatur. Als Prozessor wird der 65816 von GTE in München verwendet, der mit 1 MHz bis maximal 2,8 MHz getaktet ist (theoretisch 4 MHz). In der Grundausstattung umfaßt der Apple GS 128K ROM und 256K RAM. Als Schnittstellen gibt es neben externen Ports (wie beim IIc) zusätzlich interne Slots (wie beim IIe), und zwar 7 normale Slots (hinten) und einen RAM-Karte-Slot (vorne). Ferner ist eine Uhr mit Batterie eingebaut und ein Sockel für einen Synthesizer-Chip reserviert. Die Maus wird serienmäßig mitgeliefert. Der Anschluß von Farbmonitoren ist möglich. Sonstige Peripheriegeräte (Diskettenlaufwerke mit 140K im 5,25-Zoll-Format oder mit 800K im 3,5-Zoll-Format, Festplatte HD20SC, Drucker Imagewriter I oder II usw.) sind gesondert erhältlich. Das alte 8-Bit-Applesoft-BASIC befindet sich noch im ROM. 16-Bit-Programmiersprachen gibt es dagegen zur Zeit noch nicht. Als 16-Bit-Betriebssystem soll ein neues ProDOS im Lieferumfang enthalten sein.

Eine typische Grundkonfiguration (Apple GS mit Tastatur und Maus, mit 256K RAM und Applesoft im 128K-ROM, mit Schwarzweißmonitor und externem 800K-Drive) kostet als Paketpreis ca. DM 4500,–. Als vordringliche Erweiterungen kämen dann noch die Kosten für einen Drucker sowie für ein zweites Laufwerk oder ggf. eine Festplatte hinzu.

Atari ST

Der Atari ST wird in einer Reihe von Ausbaustufen geliefert. Die neueste Version vom Frühjahr 1986 ist der 1040 ST. Bei diesem Gerät ist die Tastatur und ein 720K-Diskettenlaufwerk im 3,5-Zoll-Format in die Konsole integriert. Als Prozessor wird der 68000 von Motorola verwendet, der mit 7,9 MHz getaktet ist. In der Grundausstattung umfaßt der 1040 ST 192K ROM und 1024K RAM. Als Schnittstellen gibt es nur externe Ports, und zwar u.a. für Monitor, Diskettenlaufwerk, Festplatte, Drucker (meist Parallelschnittstelle), Modem oder Drucker (V.24 bzw. RS232C), Synthesizer (MIDI-Ein- und -Ausgänge), Maus und Joystick. Die Maus wird serienmäßig mitgeliefert. Ferner gibt es einen nach außen geführten Steckplatz für ROM-Module. Der Anschluß von Farbmonitoren ist möglich. Sonstige Peripheriegeräte (zweites Diskettenlaufwerk mit wahlweise 360K oder 720K, Festplatte mit 20 Megabytes, Drucker usw.) sind gesondert erhältlich. Logo und BASIC werden als 16-Bit-Programmiersprachen auf Diskette mitgeliefert.

Eine typische Grundkonfiguration (Atari 1040 ST mit Maus und Tastatur, mit 1024K RAM, Betriebssystem im 192K-ROM, eingebautem 720K-Drive und externem Schwarzweiß-Monitor sowie mit Logo und BASIC) kostet als Paketpreis ca. DM 3300,–. Als vordringliche Erweiterungen kämen dann noch die Kosten für einen Drucker sowie für ein zweites Laufwerk oder ggf. eine Festplatte hinzu.

Das nach wie vor lieferbare Alternativmodell 520 ST+ vom Herbst 1985 umfaßt zwar ebenfalls 1024K RAM, doch befindet sich der Großteil des Betriebssystems (TOS) nicht im ROM, sondern muß von Diskette geladen werden. Ferner ist das Diskettenlaufwerk nicht in die Konsole integriert, sondern wird extern angeschlossen. Ansonsten gibt es keine wesentlichen Unterschiede. Die älteren Geräte (Atari 260 ST), die nur über 512K RAM verfügen, sind noch erhältlich.

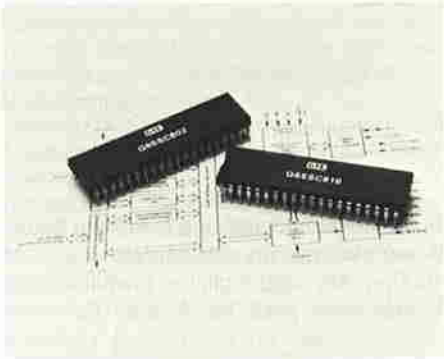


Abb. 1 G65SC816 (Quelle: GTE, München)

3. Prozessor

Die Leistung von Prozessoren wird oft überbewertet, denn Programme werden nicht von Prozessoren, sondern von Programmierern gemacht. Das folgende BASIC-Programm mag diese These erhärten:

```
FOR X = 1 TO 10000
 Y = SIN (X)
NEXT X
```

Für die Aufgabe benötigen:

Macintosh	5,5 MHz	270s	8 Bytes
Apple IIe	1 MHz	265s	5 Bytes
IBM-PC	4,7 MHz	202s	4 Bytes
Apple GS	2,8 MHz	90s	5 Bytes
Apple CP/M	6 MHz	82s	4 Bytes
Apple IIe	3,3 MHz	80s	5 Bytes
Atari	8 MHz	14s	6 Bytes

Wenn man ein derartiges Programm auf verschiedenen Rechnern laufen läßt, muß man nicht nur den Prozessor und seine Taktfrequenz, sondern auch die Art (hier binäre statt BCD-Arithmetik) und Genauigkeit (hier Mantisse mit 6 bis 16 Stellen) der Berechnung ins Kalkül ziehen. Trotzdem wird man seine Überraschungen erleben:

– Der alte Apple IIe mit 1-MHz-6502 benötigt für diese Aufgabe bei 9stelliger Mantisse (intern 5 Bytes) 265 Sekunden. Unter Verwendung einer Accelerator IIe mit 3,3 MHz reduziert sich die Rechenzeit dann auf 80 Sekunden. Der neue Apple GS dürfte ca. 90 Sekunden benötigen.

– Auch CP/M-MBASIC bringt es auf einem Apple II mit einer 6-MHz-Z80-Karte auf einen ähnlichen Wert, nämlich 82 Sekunden, wobei man allerdings die nur 6stellige Mantisse (intern 4 Bytes) berücksichtigen muß.

– Als ich dasselbe Programm dann jedoch unter Microsoft-BASIC auf dem Macintosh laufen ließ, traute ich meinen Augen kaum, als ich eine Zeit von über 270 Sekunden gemessen hatte (vgl. Peeker, 1/85, S. 74). Denn selbst, wenn man die 16stellige Mantisse von MBASIC sowie die Taktfrequenz von nur 5,5 MHz beim Mac-68000 berücksichtigt, ist dieses Resultat kaum verständlich. (Übrigens braucht auch das IBM-PC-BASIC beim 4,7-MHz-8086 und bei 6stelliger Mantisse über 200 Sekunden.)

– Daß es auch anders geht, zeigt das Atari-BASIC, das die Aufgabe in 25 Sekunden erledigt (6stellige Mantisse, intern 4 Bytes, 7,9-MHz-68000-CPU).

– Und daß es noch viel schneller und besser geht, zeigt das neue GFA-BASIC für den Atari St, das bei einer 11stelligen Mantisse (intern 6 Bytes) nur 14 Sekunden benötigt.

Der 68000 ist im allgemeinen dem 65816 überlegen, wenn man die Befehlsstruktur zugrunde legt. So verfügt der 68000 über mehr als ein Dutzend 32-Bit-Register und entsprechende 32-Bit-Befehle, was bei dem 65816 nicht der Fall ist.

```
Rem Primzahlen nach Eratosthenes
Rem ca. 2,3s bis 8191
T1=Timer
H%=8191
Dim P%(H%)
S%=Int(Sqr(H%))+1
T%=-1
F%=0
Rem 10mal durchlaufen
For Loop%=1 To 10
  Arrayfill P%(),T%
  For X%=2 To S%
    If P%(X%)=T% Then
      If X%+X%<H% Then
        For Y%=X%+X% To H% Step X%
          P%(Y%)=F%
        Next Y%
      Endif
    Endif
  Next X%
Next Loop%
T2=Timer
Print (T2-T1)/200;" Sekunden"
```

Abb. 2 GFA-BASIC-Primzahlenprogramm

Ein Kompaktkurs zu diesem pascalähnlichen Atari-BASIC beginnt im nächsten Peeker.

Schwieriger ist die Beantwortung der Frage, wie das Taktfrequenz-Verhältnis zwischen 68000 und 65816 sein müßte, damit der 65816 dem 68000 zumindest ebenbürtig ist, denn der 65816 braucht meist weniger als 6 Takte pro Befehl, während der 68000 für einige Befehle, z.B. DIVU und DIVS, über 140 Takte benötigt. Solche 68000-Befehle müssen dann allerdings beim 65816 durch ganze Befehlssequenzen simuliert werden.

Ich möchte an dieser Stelle die These aufstellen, daß ein x-MHz-65816 ungefähr einem 2x-MHz-68000 entspricht (z.B. 4-MHz-65816 versus 8-MHz-68000). Dies soll zu einem späteren Zeitpunkt im Zusammenhang mit einem 65816-Kompaktkurs (vgl. 68000-Kompaktkurs im September-Heft) näher begründet werden.

Der 65816 im Apple GS könnte theoretisch mit 4 MHz laufen, ist jedoch aus Gründen, die uns niemand von Apple München erklären konnte, nur mit maximal 2,8 MHz getaktet. Dies ist weniger als bei den Beschleunigungskarten für den Apple IIe (Accelerator, Speedemon, Transwarp u.a.), deren 65C02C-CPUs mit maximal 3,3 MHz getaktet sind. Damit ist der alte Apple IIe mit Accelerator IIe schneller als der neue Apple GS im 8-Bit-Modus. Eine enttäuschende Erkenntnis!

Der Atari ST läuft ohne Wartezyklen mit den vollen 8 MHz (genauer ca. 7,9 MHz), während beispielsweise der Macintosh im RAM mit heruntergetakteten 5,5 MHz (und im ROM mit vollen 7,9 MHz) gefahren wird. Damit ist der Atari eindeutig schneller als der Macintosh. Ferner ist der Atari auch eindeutig schneller als der Apple GS. Umgekehrt ist nach unserer obigen x-zu-2x-These der 2,8-MHz-65816 des Apple GS dem 5,5-MHz-68000 des Macintosh weitgehend ebenbürtig.

Technisch sei an dieser Stelle nachgetragen, daß der 65816 des Apple GS softwaremäßig zwischen 8- und 16-Bit-Modus und zwischen 1- und 2,8-MHz-Taktfrequenz umgeschaltet werden kann. Im Monitor finden sich beispielsweise Befehlssequenzen in der Art

```
CLC              ;16-Bit-Modus
XCE
LDA $C036
ORA #%10000000   ;Fast-Modus
STA $C036
```

Die ROM-Routinen werden im Fast-Modus und mit Ausnahme des alten Applesoft-BASIC auch im 16-Bit-Modus ausgeführt. Ferner funktioniert das Hoch- und Heruntertakten auch beim Diskettenzugriff bei den alten 6502-Betriebssystemen (DOS 3.3, ProDOS, UCSD-Pascal; mög-

licherweise auch CP/M; letzteres nicht ausprobiert!). Daraus folgt, daß man wie bei der Accelerator-Karte Altprogramme im Fast-Modus fahren kann.

4. RAM

Der 65816 und der 68000 sind beide in der Lage, 24-Bit-Adressen, also 2 hoch 24 = 16 Megabytes anzusprechen. Die Atari-Computer 1040 ST und 520 ST+ werden serienmäßig mit 1 Megabyte RAM ausgeliefert, während der Apple GS serienmäßig nur über 256K RAM verfügt. Mit einer entsprechenden RAM-Karte, die in den achten Slot gesteckt wird, der sich übrigens nicht in der hinteren Slot-Leiste, sondern vorne befindet, ist eine Erweiterung auf 1 Megabyte und mehr ohne weiteres möglich. Doch kennen wir inzwischen die Add-on-Preispolitik der Firma Apple, so daß für eine 1-M-RAM-Karte schätzungsweise DM 2000,– verlangt werden wird, während beispielsweise eine Erweiterung des Atari 1040 ST auf 3 Megabytes nur DM 1300,– und die Aufrüstung des alten Atari 260 ST von 512K auf 1024K nur DM 250,– kosten.

5. ROM

Der Atari 1040 ST enthält im 192K-ROM den größten Teil des Betriebssystems, das bei den Vormodellen als TOS-Image (TOS = Tramiel Operating System) von Diskette geladen werden mußte und damit bei den älteren 512K-Geräten wertvollen Speicherraum in Beschlag nahm. Gleiches gilt für den alten Macintosh im Gegensatz zum Macintosh Plus, der nun mit einem 128K-ROM aufwartet.
Bei Apple GS befinden sich im 128K-ROM nicht nur das alte Applesoft-BASIC sowie der (stark gepatchte) Monitor, sondern darüber hinaus auch Schnittstellentreiber, Speicherverwaltungsroutinen für RAM-Karten und ein Konfigurierungsprogramm.

6. Tastatur

Der Atari 1040 ST hat eine eingebaute und der Apple GS eine externe Tastatur. Beide Tastaturen sind recht flach und lassen kaum Wünsche offen, denn sie verfügen zusätzlich über Cursortasten (beim Atari als gesonderter Cursorblock) sowie über einen Ziffernblock. Die Atari-ST-Tasten vermitteln ein federndes, die Apple-GS-Tasten ein klackendes Gefühl. Im übrigen sind Tastaturen in meinen Augen Geschmacks- und Gewöhnungssache. Es ist zudem bei beiden Geräten möglich, Fremdtastaturen anzuschließen. Wichtig ist, daß beim Atari 1040 ST 256 verschiedene Codes erzeugt werden können, so daß der IBM-ähnliche Atari-Zeichensatz auch 256 verschiedene Zeichen umfaßt, während die Apple-GS-Tastatur offenbar wie beim alten Apple II nur 128 verschiedene Codes generieren kann.

7. Bildschirm

Hatte der Macintosh bereits einen gestochen scharfen Bildschirm – allerdings nur mit einer 9-Zoll-Diagonale, so ist der 12-Zoll-Atari-Monitor noch eine Nuance schärfer. Die Auflösung beträgt beim Atari-Monitor 640 Bildpunkte in der Breite und 400 Bildpunkte in der Höhe. Bei einer 8x16-Punkt-Matrix können 80 Zeichen in der Breite (640 : 8 = 80) und 25 Zeilen in der Höhe (400 : 16 = 25) dargestellt werden. Dies gilt für die normale Schrift des Schwarzweißmonitors. Beim dem um DM 400,– teureren Farbmonitor, der allerdings nur für Spezialanwendungen und Spielprogramme empfohlen wird, reduziert sich die Auflösung entsprechend (320 mal 200 Bildpunkte mit 16 Farben, 640 mal 200 Bildpukte mit 4 Farben). Übrigens ist die Auflösung beim Macintosh mit 512 Bildpunkten in der Breite und 342 Bildpunkten in der Höhe geringer als beim Atari ST. Mißt man beim Macintosh die Anzahl der Punkte pro cm nicht aufgrund des Bildschirms (= Fläche der Glasscheibe), sondern aufgrund der Bildfläche (= genutzte Fläche innerhalb des Bildschirms), so kommt man beim Macintosh auf einen 28er Raster (= 28 Punkte pro cm) gegenüber einem 30er Raster beim Atari ST. Der geringe Unterschied rührt daher, daß der Atari-Monitor größer als der Mac-Monitor ist. Würden beide Geräte einen 12-Zoll-Monitor mit gleicher Bildfläche benutzen, so hätte der Atari ST eine deutlich höhere Rasterweite (ca. 35er Raster).
Für drucktechnisch Interessierte sei darauf hingewiesen, daß Tageszeitungen für Abbildungen einen 25er bis 35er Raster verwenden, wohingegen für normale Abbildungen auf Offsetpapier ein 60er bis 70er Raster üblich ist. Dies gilt jedoch nicht für die Lichtsatzschriften, bei denen die Auflösung mindestens einem 300er bis 400er Raster, bei Qualitätsdrucksachen sogar einem 2048er Raster entspricht.

Die Schriftqualität des Apple GS ist eine herbe Enttäuschung. Sie ist nämlich mit der des IIe identisch und damit technisch völlig überholt. Von den in diesem Jahr neu herausgebrachten Mikrocomputern dürfte nur noch der Apple GS diese bescheidene Auflösung haben. Zum Programmieren habe ich beim Apple IIe regelmäßig den 40-Z/Z-Modus eingeschaltet, um die Augen zu schonen. Andere haben sich eine Ultraterm gekauft, die ein sehr gutes Schriftbild liefert. Nunmehr mutet man uns wieder im 80-Z/Z-Modus eine 7x8-Punkt-Matrix zu, bei der die waagrechten Punktreihen verschmelzen, während die senkrechten Punktreihen getrennt sind, womit die Schrift keinen homogenen, augenfreundlichen Helligkeitswert hat. Was die Apple-GS-Schrift so schlecht macht, ist nämlich nicht nur die geringe Matrix, sondern auch die Tatsache, daß die Bildpunkte nicht immer die gleiche Punktfläche einnehmen.
Neben den für Apple-IIe/c-Besitzer bereits bekannten Hires-Auflösungen (Hires 280x192 mit 6 Farben, Double Hires

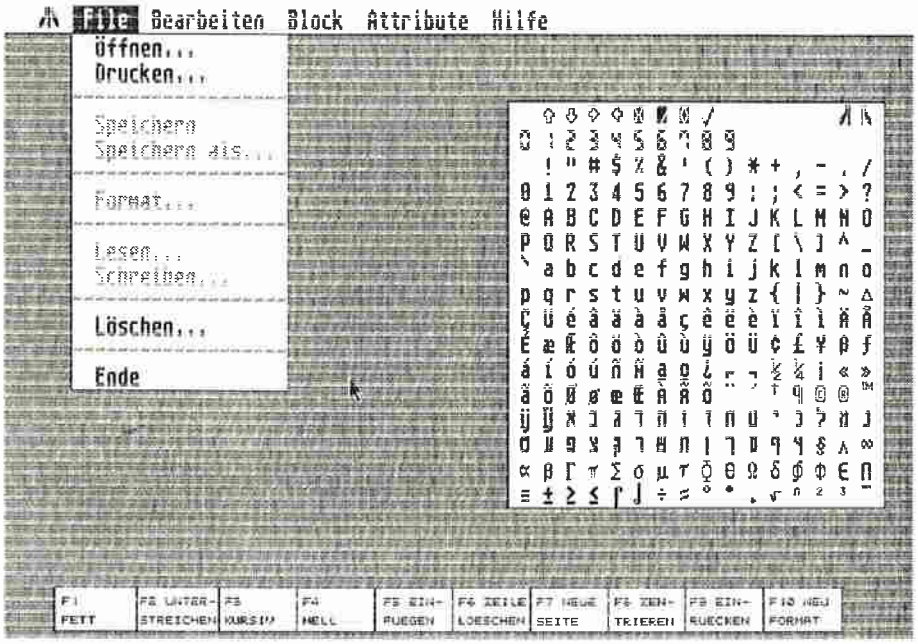


Abb. 3 Atari-Zeichensatz

560x192 mit 16 Farben) soll es auch eine Super Hires (320x200 mit 16 Farben und 640x200 mit 4 Farben) geben. Wie man diese erzeugt, ist bislang unklar, denn als der Apple GS von uns im August zum zweitenmal in München einen Tag lang getestet wurde, waren 16-Bit-Programme noch nicht verfügbar.

Um Mißverständnissen vorzubeugen, sei abschließend darauf hingewiesen, daß der Apple GS wie die früheren Apple-II-Typen über einen getrennten Grafik- und Textmodus und sinngemäß über getrennte Grafik- und Text-Speicher im RAM verfügt. Anders liegen die Dinge beim Macintosh und Atari ST, die beide nur noch einen Grafikmodus haben. Dabei gibt es im einzelnen folgende Unterschiede:

– Der Atari-Grafikbildschirm, der übrigens keinen absoluten RAM-Bereich einnimmt, sondern sich in der Regel am Ende des jeweiligen RAMs (512K, 1024K usw.) befindet, belegt ca. 32K, während der Macintosh-Grafikbildschirm ca. 22K des RAM-Speichers beansprucht.

– Beim Macintosh ist die Fenster- und Maustechnik obligatorisch und beim Atari ST optional. Insbesondere die Programmiersprachen machen aus Gründen der Geschwindigkeit von der Fenstertechnik nur sparsamen oder oft gar keinen Gebrauch (Beispiele: GFA-BASIC, UCSD-Pascal). Dann merkt man gar nicht, daß man sich im Grafikmodus befindet.

– Noch wichtiger ist jedoch die Tatsache, daß der Atari ST in der Regel Schriftzeichen mit gleichen Dickten (= Buchstabenbreiten) benutzt, während der Macintosh fast immer Schriftzeichen mit unterschiedlichen Dickten verwendet. So sind etwa die folgenden zwei Zeilen

WWWWWWWWWW
vvvvvvvvvvvvvvvvvvv

beim Macintosh etwa gleichlang, während beim Atari ST die zweite Zeile doppelt so lang wie die erste ist. Daraus folgt, daß die Schriftzeichen des Atari ST den Schreibmaschinentypen und die des Macintosh den Drucktypen ähneln. Die zweifellos schöneren Macintosh-Schriften werden jedoch mit einem starken Geschwindigkeitsverlust bei der Bildschirmsteuerung erkauft. Betrachten wir hierzu folgendes BASIC-Programm, das 1000 Strings mit einer Länge von je 55 Zeichen ausgibt:

```
FOR X = 1000 TO 2000
PRINT X; "-1234567890123456789 0
 1234567890123456789 0123456789 0"
NEXT
```

Die Ausführung dieses Programms beträgt beim Atari ST unter GFA-BASIC ca. 50 Sekunden, beim Apple GS unter Ap-

ADC	Add Memory to Accumulator with Carry	PHA	Push Accumulator on Stack
AND	"AND" Memory with Accumulator	PHB	Push Data Bank Register on Stack
ASL	Shift One Bit Left, Memory or Accumulator	PHD	Push Direct Register on Stack
BCC*	Branch on Carry Clear (Pc = 0)	PHK	Push Program Bank Register on Stack
BCS*	Branch on Carry Set (Pc = 1)	PHP	Push Processor Status on Stack
BEQ	Branch if Equal (Pz = 1)	PHX	Push Index X on Stack
BIT	Bit Test	PHY	Push Index Y on Stack
BMI	Branch if Result Minus (PN = 1)	PLA	Pull Accumulator from Stack
BNE	Branch if Not Equal (Pz = 0)	PLB	Pull Data Bank Register from Stack
BPL	Branch if Result Plus (PN = 0)	PLD	Pull Direct Register from Stack
BRA	Branch Always	PLP	Pull Processor Status from Stack
BRK	Force Break	PLX	Pull Index X from Stack
BRL	Branch Always Long	PLY	Pull Index Y form Stack
BVC	Branch on Overflow Clear (Pv = 0)	REP	Reset Status Bits
BVS	Branch on Overflow Set (Pv = 1)	ROL	Rotate One Bit Left (Memory or Accumulator)
CLC	Clear Carry Flag	ROR	Rotate One Bit Right (Memory or Accumulator)
CLD	Clear Decimal Mode	RTI	Return from Interrupt
CLI	Clear Interrupt Disable Bit	RTL	Return from Subroutine Long
CLV	Clear Overflow Flag	RTS	Return from Subroutine
CMP*	Compare Memory and Accumulator	SBC	Subtract Memory from Accumulator with Borrow
COP	Coprocessor	SEC	Set Carry Flag
CPX	Compare Memory and Index X	SED	Set Decimal Mode
CPY	Compare Memory and Index Y	SEI	Set Interrupt Disable Status
DEC*	Decrement Memory or Accumulator by One	SEP	Set Processor Status Bite
DEX	Decrement Index X by One	STA	Store Accumulator in Memory
DEY	Decrement Index Y by One	STP	Stop the Clock
EOR	"Exclusive OR" Memory with Accumulator	STX	Store Index X in Memory
INC*	Increment Memory or Accumulator by One	STY	Store Index Y in Memory
INX	Increment Index X by One	STZ	Store Zero in Memory
INY	Increment Index Y by One	TAX	Transfer Accumulator to Index X
JML**	Jump Long	TAY	Transfer Accumulator to Index Y
JMP	Jump to New Location	TCD*	Transfer Accumulator to Direct Register
JSL**	Jump Subroutine Long	TCS*	Transfer Accumulator to Stack Pointer Register
JSR	Jump to New Location Saving Return Address	TDC*	Transfer Direct Register to Accumulator
LDA	Load Accumulator with Memory	TRB	Test and Reset Bit
LDX	Load Index X with Memory	TSB	Test and Set Bit
LDY	Load Index Y with Memory	TSC*	Transfer Stack Pointer Register to Accumulator
LSR	Shift One Bit Right (Memory or Accumulator)	TSX	Transfer Stack Pointer Register to Index X
MVN	Block Move Negative	TXA	Transfer Index X to Accumulator
MVP	Block Move Positive	TXS	Transfer Index X to Stack Pointer Register
NOP	No Operation	TXY	Transfer Index X to Index Y
ORA	"OR" Memory with Accumulator	TYA	Transfer Index Y to Accumulator
PEA	Push Effective Absolute Address on Stack (or Push Immediate Data on Stack)	TYX	Transfer Index Y to Index X
		WAI	Wait for Interrupt
PEI	Push Effective Indirect Address on Stack (add one cycle if DL ≠ 0)	XBA*	Exchange B and A Accumulator
		XCE	Exchange Carry and Emulation Bits
PER	Push Effective Program Counter Relative Address on Stack		

Abb. 4 G65SC816-Befehlssatz (Quelle: GTE, München)

MSD	LSD																MSD
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRK s 2 8	ORA (d,x) 2 6	COP s 2 8	ORA r,s 2 4	TSB d 2 5	ORA d 2 3	ASL d 2 5	ORA [d] 2 6	PHP s 1 3	ORA # 2 2	ASL A 1 2	PHD s 1 4	TSB a 3 6	ORA a 3 4	ASL a 3 6	ORA al 4 5	0
1	BPL r 2 2	ORA (d),y 2 5	ORA (d) 2 5	ORA (r,s),y 2 7	TRB d 2 5	ORA d,x 2 4	ASL d,x 2 6	ORA [d],y 2 6	CLC i 1 2	ORA a,y 3 4	INC A 1 2	TCS i 1 2	TRB a 3 6	ORA a,x 3 4	ASL a,x 3 7	ORA al,x 4 5	1
2	JSR a 3 6	AND (d,x) 2 6	JSL al 4 8	AND r,s 2 4	BIT d 2 3	AND d 2 3	ROL d 2 5	AND [d] 2 6	PLP s 1 4	AND # 2 2	ROL A 1 2	PLD s 1 5	BIT a 3 4	AND a 3 4	ROL a 3 6	AND al 4 5	2
3	BMI r 2 2	AND (d),y 2 5	AND (d) 2 5	AND (r,s),y 2 7	BIT d,x 2 4	AND d,x 2 4	ROL d,x 2 6	AND [d],y 2 6	SEC i 1 2	AND a,y 3 4	DEC A 1 2	TSC i 1 2	BIT a,x 3 4	AND a,x 3 4	ROL a,x 3 7	AND al,x 4 5	3
4	RTI s 1 7	EOR (d,x) 2 6	reserve 2 2	EOR r,s 2 4	MVP xya 3 7	EOR d 2 3	LSR d 2 5	EOR [d] 2 6	PHA s 1 3	EOR # 2 2	LSR A 1 2	PHK s 1 3	JMP a 3 3	EOR a 3 4	LSR a 3 6	EOR al 4 5	4
5	BVC r 2 2	EOR (d),y 2 5	EOR (d) 2 5	EOR (r,s),y 2 7	MVN xya 3 7	EOR d,x 2 4	LSR d,x 2 6	EOR [d],y 2 6	CLI i 1 2	EOR a,y 3 4	PHY s 1 3	TCD i 1 2	JMP al 4 4	EOR a,x 3 4	LSR a,x 3 7	EOR al,x 4 5	5
6	RTS s 1 6	ADC (d,x) 2 6	PER s 3 6	ADC r,s 2 4	STZ d 2 3	ADC d 2 3	ROR d 2 5	ADC [d] 2 6	PLA s 1 4	ADC # 2 2	ROR A 1 2	RTL s 1 6	JMP (a) 3 5	ADC a 3 4	ROR a 3 6	ADC al 4 5	6
7	BVS r 2 2	ADC (d),y 2 5	ADC (d) 2 5	ADC (r,s),y 2 7	STZ d,x 2 4	ADC d,x 2 4	ROR d,x 2 6	ADC [d],y 2 6	SEI i 1 2	ADC a,y 3 4	PLY s 1 4	TDC i 1 2	JMP (a,x) 3 6	ADC a,x 3 4	ROR a,x 3 7	ADC al,x 4 5	7
8	BRA r 2 2	STA (d,x) 2 6	BRL rl 3 3	STA r,s 2 4	STY d 2 3	STA d 2 3	STX d 2 3	STA [d] 2 6	DEY i 1 2	BIT # 2 2	TXA i 1 2	PHB s 1 3	STY a 3 4	STA a 3 4	STX a 3 4	STA al 4 5	8
9	BCC r 2 2	STA (d),y 2 6	STA (d) 2 5	STA (r,s),y 2 7	STY d,x 2 4	STA d,x 2 4	STX d,y 2 4	STA [d],y 2 6	TYA i 1 2	STA a,y 3 5	TXS i 1 2	TXY i 1 2	STZ a 3 4	STA a,x 3 5	STZ a,x 3 5	STA al,x 4 5	9
A	LDY # 2 2	LDA (d,x) 2 6	LDX # 2 2	LDA r,s 2 4	LDY d 2 3	LDA d 2 3	LDX d 2 3	LDA [d] 2 6	TAY i 1 2	LDA # 2 2	TAX i 1 2	PLB s 1 4	LDY a 3 4	LDA a 3 4	LDX a 3 4	LDA al 4 5	A
B	BCS r 2 2	LDA (d),y 2 5	LDA (d) 2 5	LDA (r,s),y 2 7	LDY d,x 2 4	LDA d,x 2 4	LDX d,y 2 4	LDA [d],y 2 6	CLV i 1 2	LDA a,y 3 4	TSX i 1 2	TYX i 1 2	LDY a,x 3 4	LDA a,x 3 4	LDX a,y 3 4	LDA al,x 4 5	B
C	CPY # 2 2	CMP (d,x) 2 6	REP # 2 3	CMP r,s 2 4	CPY d 2 3	CMP d 2 3	DEC d 2 5	CMP [d] 2 6	INY i 1 2	CMP # 2 2	DEX i 1 2	WAI i 1 3	CPY a 3 4	CMP a 3 4	DEC a 3 6	CMP al 4 5	C
D	BNE r 2 2	CMP (d),y 2 5	CMP (d) 2 5	CMP (r,s),y 2 7	PEI s 2 6	CMP d,x 2 4	DEC d,x 2 6	CMP [d],y 2 6	CLD i 1 2	CMP a,y 3 4	PHX s 1 3	STP i 1 3	JML (a) 3 6	CMP a,x 3 4	DEC a,x 3 7	CMP al,x 4 5	D
E	CPX # 2 2	SBC (d,x) 2 6	SEP # 2 3	SBC r,s 2 4	CPX d 2 3	SBC d 2 3	INC d 2 5	SBC [d] 2 6	INX i 1 2	SBC # 2 2	NOP i 1 2	XBA i 1 3	CPX a 3 4	SBC a 3 4	INC a 3 6	SBC al 4 5	E
F	BEQ r 2 2	SBC (d),y 2 5	SBC (d) 2 5	SBC (r,s),y 2 7	PEA s 3 5	SBC d,x 2 4	INC d,x 2 6	SBC [d],y 2 6	SED i 1 2	SBC a,y 3 4	PLX s 1 4	XCE i 1 2	JSR (a,x) 3 6	SBC a,x 3 4	INC a,x 3 7	SBC al,x 4 5	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

symbol	addressing mode	symbol	addressing mode
#	immediate	[d]	direct indirect long
A	accumulator	[d],y	direct indirect indexed long
r	program counter relative	a	absolute
rl	program counter relative long	a,x	absolute indexed (with x)
i	implied	a,y	absolute indexed (with y)
s	stack	al	absolute long
d	direct	al,x	absolute indexed long
d,x	direct indexed (with x)	r,s	stack relative
d,y	direct indexed (with y)	(r,s),y	stack relative indirect indexed
(d)	direct indirect	(a)	absolute indirect
(d,x)	direct indexed indirect	(a,x)	absolute indexed indirect
(d),y	direct indirect indexed	xya	block move

legend	
instruction mnemonic	addressing mode
base number of bytes	base number of cycles

Abb. 5 G65SC816 Opcodes (Quelle: GTE, München)

plesoft-BASIC schätzungsweise ca. 55 Sekunden (noch nicht nachgemessen!) und beim Macintosh unter Microsoft-BASIC ca. 245 Sekunden. Daraus folgt: Die Bildschirmsteuerung ist beim Atari ST im Grafikmodus mindestens ebenso schnell wie beim Apple GS im Textmodus, während der Macintosh wegen seiner Proportionalschriften weit abgeschlagen ist.

8. Konfiguration

Der Apple GS verfügt über ein Konfigurationsprogramm (Control Panel Program), mit dessen Hilfe bestimmte Systemkonstanten festgelegt werden können. Dieses ROM-residente Programm wird durch die Tastenkombination Option-Ctrl-Reset ausgelöst. (Die vom Macintosh her bekannte Option-Taste entspricht der geschlossenen bzw. schwarzen Apfel-Taste des Apple IIe/c). Danach können im einzelnen eingestellt werden:

– Bildschirmdarstellung (40 Z/Z oder 80 Z/Z, Farben für Schrift-, Hintergrund- und Bildflächenumrandung)
– Lautsprecher (Lautstärke und Tonhöhe)
– Uhr (Datum und Uhrzeit)
– Taktfrequenz (1 MHz oder 2,8 MHz)
– Tastatur (Tastenbelegung: deutsch, englisch u.a., Wiederhol-Geschwindigkeit und -Verzögerung, Cursor-Blinkgeschwindigkeit)
– Slot-Belegung (Da externer Port und entsprechender interner Slot nicht gleichzeitig benutzt werden können, gibt man hier an, was aktiv sein soll. Ferner kann man virtuelle Slots und Boot-Slots festlegen.)
– Drucker- und Modemkonfigurierung (Baudrate, Parität, XON/XOFF-Protokoll, Bildschirmecho u.ä.)

Die definierten Konstanten bleiben auch nach dem Ausschalten des Apple GS batteriegepuffert im RAM (der Uhr?). Sie können jederzeit durch erneuten Aufruf des Konfigurationsprogramms sowie teilweise auch von Applesoft-BASIC aus durch entsprechende Pokes geändert werden. Beispielsweise kann man über den Softswitch $C036 die Taktfrequenz und über die Softswitches $C022 und $C034 die Bildschirmfarben kontrollieren.

Beim Atari ST werden ähnliche Konfigurierungen wie beim Apple GS über die GEM-Menüleiste „Desk" vorgenommen, z.B. Modem-Einstellung, Drucker-Anpassung u.a.m. Über die Option Kontrollfeld im engeren Sinne (entspricht in etwa dem Kontrollfeld des Macintosh) werden Tastaturkonstanten (Wiederhol-Geschwindigkeit und -Verzögerung, Tastenklick, Datum und Uhrzeit sowie für den Farbmonitor die Farbpalette festgelegt. Der Atari ST hat übrigens standardmäßig keine Uhr, sie kann aber für ca. DM 100,– gesondert erworben werden.

Die Systemkonstanten (Desktop-Info usw.) werden auf Diskette gespeichert (über Menüoption „Arbeit sichern"). Dies ist insofern flexibler als beim Apple GS, weil zusätzlich eigene Konfigurierungsroutinen und Desk-Accessories (Drucker-Spooler, Taschenrechner usw.) entwickelt werden können. Unter der neuen 8-Bit-ProDOS-Version 1.2 sowie der 16-Bit-ProDOS-Version sollen allerdings ebenfalls Desk-Accessories installierbar sein.

9. Betriebssystem

Über die Betriebssysteme der beiden Rechner werden im Peeker gesonderte Beiträge erscheinen, so daß wir uns an dieser Stelle mit einigen kurzen Bemerkungen begnügen können.

Das Hauptbetriebssystem des Apple GS wird „ProDOS 16" heißen und wahrscheinlich nur auf 3,5-Zoll-Diskette geliefert werden. Dieses hierarchische Betriebssystem, das Mitte August bei Apple München noch nicht vorlag, wird weitgehend dem alten ProDOS 1.1.1 entsprechen, welches übrigens durch ProDOS 1.2 ersetzt wird. Allerdings dürften für die 16-Bit-Version die Parameter erweitert worden sein, da man jetzt Binärdateien mit einer Länge von mehr als 64K einladen können muß. Ob man mit 8-Bit-ProDOS auf Disketten zugreifen kann, die unter 16-Bit-ProDOS erzeugt wurden und umgekehrt, ist bislang noch unbekannt.

Neben ProDOS können alle alten 8-Bit-Betriebssysteme (DOS 3.3, Diversi-DOS, Pascal 1.1 und 1.2) mit 5,25-Zoll-Disketten (140K) oder auch mit 3,5-Zoll-Disketten (Pascal 1.3) verwendet werden. Darüber hinaus soll auch CP/M in Verbindung mit einer entsprechenden Z80-Karte lauffähig sein. „Entsprechend" heißt, daß die bisherigen CP/M-Karten möglicherweise nicht funktionieren, wie wir von dritter Seite erfahren haben.

Beim Atari ST muß man zwischen dem Betriebssystem im engeren Sinne (TOS-Kernel = TOS-Kern) und der Benutzerschnittstelle (GEM-Shell = GEM-Schale) unterscheiden. Der TOS-Kernel ist wie der ProDOS-Kernel ein hierarchisches, unixähnliches Betriebssystem mit Volume-Directory, Subdirectories, Subsubdirectories usw., das eine gewisse Ähnlichkeit mit MS-DOS nicht leugnen kann.

Als Benutzerschnittstelle gibt es für Programmierer beispielsweise das schlichte Command-Programm (DOS-Shell) ohne Fenster und Maus, mit dessen Hilfe Befehle wie DIR (directory) CD (change directory), TYPE (CP/M-ähnlicher Dump-Befehl) usw. eingegeben werden können. Demgegenüber wird als Benutzerschnittstelle für Anwender normalerweise das GEM-Desktop verwendet, das im großen und ganzen dem Macintosh-Desktop entspricht. Hier wird dann beispielsweise ein Subdirectory names ORDNER nicht mit RMDIR ORDNER gelöscht, sondern dadurch, daß das Bild des Ordners mit der Beschriftung „ORDNER" mittels der Maus über das Bild des Papierkorbs placiert wird. Verglichen mit dem Macintosh-Desktop fällt auf, daß das Atari-Desktop für mitdenkende Benutzer gedacht ist. Beispiel: Mit dem Placieren auf den Papierkorb ist beim Atari eine Datei definitiv gelöscht, während man beim Macintosh zusätzlich noch den Papierkorb ausleeren muß. Deshalb darf man beim Macintosh eine Diskette auch nicht selbst auswerfen, sondern muß den Auswurf über eine Menü-Option herbeiführen.

10. Fazit

Apple GS

Wie man in alten Ausgaben der Zeitschrift „Byte" nachlesen kann, ist der Apple GS bereits 1984 entwickelt worden. Durch die internen Querelen zwischen John Sculley und Steve Jobs ruhte das Gerät dann zwei Jahre lang im Schrank, weil Jobs die Apple-II-Serie aus der Welt schaffen wollte.

Der wesentliche Nachteil des Apple GS besteht deshalb darin, daß er viel zu spät erscheint, denn die Apple-Landschaft hat sich verändert: Des Wartens überdrüssig, sind viele Apple-Besitzer in Richtung Atari ST und IBM-PC abgewandert. Dies gilt insbesondere für die Bundesrepublik, während die Position von Apple in den USA nach wie vor sehr stark ist.

Um den Apple GS wie den Atari ST auf 1 Megabyte RAM aufzurüsten, wird man tief in die Tasche greifen müssen. Ähnlich liegen die Dinge bei anderen Original-Zusatzgeräten. Beispielweise kostet die 20-Megabyte-Festplatte von Atari nur DM 2000,–, während man sich bei Apple nicht scheut, für die HD20 über DM 6000,– zu verlangen.

Als technische Unzulänglichkeiten haben wir insbesondere die niedrige 65816-Taktfrequenz von 2,8 MHZ sowie die bescheidene Bildschirmschrift herausgestellt. Auf der positiven Seite ist zu vermerken, daß die meisten alten Apple-II+/e/c-Programme, soweit sie seriös programmiert sind, sowie die meisten Erweiterungskarten (mit Ausnahme der für den nicht mehr vorhandenen Auxiliary Slot 3 gedachten Karten)

weiterverwendet werden können. Doch muß man bedenken, daß die Altprogramme 8-Bit-Programme sind. Um solche Programme zu fahren, benötigt man keinen 16-Bit-Rechner, sondern kann beispielsweise zur Geschwindigkeitssteigerung eine Accelerator-Karte einsetzen, die mit 3,3 MHz dann immer noch schneller als der 65816-Prozessor im 8-Bit-Modus ist. Entscheidend für den Apple GS wird es deshalb sein, in welchem Umfang echte 16-Bit-Programme erscheinen werden. Angekündigt sind 16-Bit-Versionen von ProDOS und Appleworks, doch von 16-Bit-Programmiersprachen ist seitens Apple bislang nicht die Rede. Den Apple GS empfehlen wir deshalb Altbesitzern, die mit ihrem alten Apple II+/e/c im großen und ganzen zufrieden waren und nunmehr ein Ersatz- oder Zweitgerät suchen, um darauf ihre Altprogramme weiterverwenden zu können. Wir empfehlen ihn weiterhin den Spezialanwendern, die auf die Flexibilität von Slots in bezug auf Interface-Karten für meß-, regel- und steuerungstechnische Zwecke angewiesen sind.

Atari ST

Der Atari ST hat inzwischen diejenige Position eingenommen, die der alte Apple II in seiner besten Zeit innehatte. Dies gilt namentlich für die Bundesrepublik, denn in den USA ist der Atari weniger stark verbreitet. Ein negativer Kritikpunkt beim Atari ST ist die eingebaute und damit nicht frei bewegliche Tastatur, die allerdings recht flach ist (Höhe der Leertaste 3,5 cm gegenüber 7 cm beim Apple IIe). Im Zusammenhang mit der Kompaktbauweise wird auch das Fehlen der Slots bemängelt, doch verfügt der Atari ST über so viele Schnittstellen, daß für den normalen Anwender (mit Ausnahme des Technikers) kaum noch Wünsche offenbleiben. Im übrigen kann man hier auch angenehme Überraschungen erleben. So besitze ich zu Hause noch einen alten Epson-MX-80-Drucker aus meiner Apple-Anfangszeit (Mitte 1981), wofür ein spezielles Grafik-Dump-Programm geschrieben werden mußte. Spaßeshalber schloß ich den MX-80 direkt an den Parallel-Port des Atari ST an. Und siehe da, nachdem ich über das Konfigurierungsprogramm lediglich die Anzahl der Punkte pro Zeile auf 960 umgestellt hatte, druckte der alte MX-80 sofort astreine Screen-Dumps im Grafikmodus aus. Ob dies auch beim Apple GS über den Parallel-Port möglich sein wird? Wir empfehlen den Atari ST allen Peeker-Lesern, die mit ihrem alten Apple II nicht mehr zufrieden sind und nunmehr mit einem leistungsfähigen und extrem preiswerten Gerät einen Neuanfang machen wollen. Der Entschluß zum Umsteigen dürfte Altbesitzern um so leichter fallen, als die Soft- und Hardware für den Atari ST in der Regel sehr preisgünstig ist. Man kann zwar auch UCSD- und CP/M-Programme auf dem Atari fahren, doch kommt seine Leistung erst bei richtigen 68000-Programmiersprachen zur Geltung. Ein Beispiel hierfür ist das in diesem Aufsatz mehrfach erwähnte GFA-BASIC, das weniger als 60K einnimmt – man hat dann auf dem Atari 1040 ST ca. 750K frei! – und phantastisch schnell ist.

Double-Hires-Tools

von Matthias Meyer

Zwei preisgünstige Programmpakete für doppelt-hochauflösende Grafik auf dem Apple IIc und IIe (mit 64K-Karte):

DHGR-Tool für Applesoft

Diskette und Manual, Einführungspreis DM 28,–

Diese Ampersand-Programmsammlung für Double-Hires und -Lores läuft unter Applesoft, und zwar sowohl unter DOS 3.3 als auch unter ProDOS. Unter anderen wurden folgende Befehle implementiert:

&1 und &2 wählen 1. und 2. Zeichensatz,
&CLEAR löscht die DHGR-Seite,
&COLOR= und &HCOLOR= wählen Double-Lores/Hires-Farben,
&DRAW und &XDRAW zeichnen DHGR-Shapes,
&DRAW AT zeichnet Grafikbeschriftungen (ASCII-Strings),
&GR, &HGR, &H, &TEXT, &T usw. schalten verschiedene Grafik- und Text-Modi ein,
&HLIN und &VLIN plotten waagrechte und senkrechte Double-Lores-Linien,
&HPLOT und &XHPLOT plotten DHGR-Linien,
&SCALE= und &ROT bestimmen Größe und Rotation von Shapes,
&LOAD und &SAVE laden und speichern Grafikseiten,
&HELP zeigt alle Befehle an,
und anderes mehr.

DHGR-Tool für Kyan-Pascal

Diskette und Manual, Einführungspreis DM 28,–

Das Kyan-Pascal-Tool umfaßt ähnliche Prozeduren wie die obigen Ampersand-Routinen, wobei jedoch noch einige Befehle, z. B. Procedure Swaphires, Procedure Background usw., sowie einige Datentypen, z. B. Shape, Chrset usw. zusätzlich aufgenommen worden sind.

Bei dem Kyan-Tool sind die Zeichensätze und die „Lookup"-Tabellen für die sehr schnellen Plotbefehle auf die 64K-Karte gelegt worden, und das Hauptmodul selbst befindet sich in der Bank 2 der Language-Card, ohne Kix-Reboot zu zerstören. Damit eignet sich dieses Kyan-Modul besser als andere Kyan-Grafik-Programme zur Einbindung in eigene Anwendungsprogramme.

Hüthig Software Service · Postfach 10 28 69 · 6900 Heidelberg 1

Atari-Benutzer-schnittstellen

für Anwender und Programmierer

GEM-Desktop und DOS-Shell

von Ulrich Stiehl

Unter Schnittstelle im technischen Sinne versteht man die Verbindung zwischen Rechner und Peripherie (z.B. V.24-Schnittstelle). Unter Schnittstelle im übertragenen Sinne versteht man die Interaktion (a) generell zwischen Mensch und Computer sowie (b) speziell zwischen Benutzer und Betriebssystem. Man spricht deshalb auch von Benutzerschnittstelle (User Interface) oder Betriebssystemoberfläche (im Gegensatz zum Betriebssystemkern).

Beim Umgang mit dem Atari sind zwei prinzipiell verschiedene Benutzerschnittstellen möglich:
– die dem Macintosh-Finder ähnelnde und überwiegend für Anwender gedachte Schnittstelle unter Einsatz von Maus- und Fenstertechnik (GEM-Desktop; GEM = Graphics Environment Manager). Beispiel: Eine Datei namens FILE wird gelöscht, indem die Datei-Ikone mit der Maus über die Papierkorb-Ikone placiert wird. Die Tastatur ist bei dieser sog. **Anwenderschnittstelle** oft entbehrlich.
– die den Betriebssystemen MS-DOS und CP/M ähnelnde und überwiegend für Programmierer gedachte Schnittstelle unter Einsatz der Tastatur (TOS-Command-Line-Shell; TOS = Tramiel Operating System). Beispiel: Eine Datei namens FILE wird gelöscht, indem der Befehl ERASE FILE über die Tastatur eingeben wird. Die Maus ist bei dieser sog. **Programmiererschnittstelle** meist entbehrlich.

Zwei „reinrassige" Vertreter dieser diametral entgegengesetzten Benutzerschnittstellen sollen nachfolgend vorgestellt werden. Wir dürfen dabei jedoch nicht vergessen, daß zahlreiche Mischformen denkbar sind und daß die Zuordnungen „GEM = Anwender" und „Command-Line = Programmierer" nur vorgenommen werden, um die wesentlichen Unterschiede besser herausarbeiten zu können. In der Praxis findet man sowohl Anwenderprogramme ohne GEM-Technik als auch Programmierhilfsmittel mit GEM-Technik.

Hierarchisches Betriebssystem

Vorab geben wir für Anfänger einige Erläuterungen zum hierarchischen Betriebssystems des Atari ST:

1. **Diskettenlaufwerke** und andere Datenspeicher werden mit den Buchstaben A, B, C usw. angesprochen.
2. **Disketten** und andere Datenträger können zwar einen Namen haben (z.B. DISK007), der physikalisch auf einer Diskettenspur (als erster Eintrag im Volume-Directory, s.u.) vermerkt ist, doch spielt dieser Name bei den Operationen des Betriebssystems keine Rolle, denn die Diskette als solche wird nur durch die Buchstaben A usw. angesprochen. Konkret: Wenn das eingebaute Laufwerk die Kennung A hat und eine beliebige Diskette namens DISK007 in Laufwerk A eingelegt wird, so wird diese Diskette mit A und nicht mit DISK007 angesprochen.
3. Physikalisch sind Atari-Disketten, die je nach Laufwerk als einseitige 360K- oder beidseitige 720K-Disketten verwendet werden, wie die Disketten anderer Betriebssysteme in **Spuren und Sektoren** (Blöcke) eingeteilt, und zwar in 80 Spuren auf der einen Seite und – bei 720K-Disketten – in zusätzlich 80 Spuren auf der anderen Seite der Diskette. Jede Spur enthält 9 Sektoren bzw. Blöcke zu je 512 Bytes. Dies ergibt für beidseitige Disketten 2 * 80 * 9 * 512 = 737.280 Bytes oder 737.280 : 1024 = 720K.
4. Sinn und Zweck einer Diskette ist die Speicherung von **Dateien** (Files: Programm- und Datendateien), die mit Namen angesprochen werden können. Ein **Dateiname** besteht aus dem eigentlichen Namen, der bis zu 8 Zeichen umfassen kann, sowie einer fakultativen Dateiartkennung (für Programme, Texte, Adressen usw.), die aus bis zu 3 Zeichen besteht. Beide Bestandteile werden durch einen Punkt abgegrenzt (Beispiele: PROGRAMM.BAS, ADRESSEN.BAK, TEXT007.DOC).
5. Um die Dateien über den Namen ansprechen zu können, müssen sich zusätzlich auf der Diskette **Inhaltsverzeichnisse** (Directories) befinden, die neben den Dateinamen Informationen über die Dateilänge und die physikalische Position der einzelnen Dateien enthalten.
6. Das mit dem Formatieren einer Leerdiskette angelegte Inhaltsverzeichnis ist das **Hauptinhaltsverzeichnis** (Volume-Directory). Dieses kann zwar 111 Dateinamen (und 1 Diskettennamen = 112 Einträge) aufnehmen, doch wäre ein derart umfäng-

liches Inhaltsverzeichnis sehr unübersichtlich und außerdem bei Festplatten trotzdem nicht ausreichend.
7. Deshalb können im Hauptinhaltsverzeichnis **Unterinhaltsverzeichnisse** (Subdirectories) angelegt werden, die neben den eigentlichen Dateien weitere Unterinhaltsverzeichnisse (Subsubdirectories) enthalten können, die ihrerseits neben Dateien wiederum weitere Unterinhaltsverzeichnisse (Subsubsubdirectories) aufnehmen können usw.

I. GEM-Desktop

Das GEM-Desktop wird in dem mitgelieferten Bedienungshandbuch vom 1.4.1986 – das sich allerdings *nicht* auf den neuen Atari 1040 ST bezieht! – recht genau beschrieben, so daß wir uns mit einer systematischen Aufzählung der implementierten Befehle begnügen können.

1. Allgemeines

Nach dem Booten der Systemdiskette („Language Disk") erscheint am oberen Bildschirmrand eine Pull-down-Menüleiste mit den Hauptrubriken Desk, Datei, Index und Extras. Diese lassen sich weiter untergliedern, so daß sich folgendes System ergibt:

1.0. Desk
1.1. Desktop-Info ∗
1.2. VT52-Emulator ∗
1.3. Kontrollfeld: DATE ∗
1.4. RS-232-Einstellung ∗
1.5. Drucker-Anpassung ∗

2.0. Datei
2.1. Öffne: DIR, RUN, TYPE
2.2. Zeige Info: VOL, REN
2.3. Neuer Ordner: MKDIR
2.4. Schließe: CD ..
2.5. Schließe Fenster: CD \
2.6. Formatiere: FORMAT

3.0. Index: DIR
3.1. als Bilder
3.2. als Text
3.3. Ordne Namen
3.4. Ordne Datum
3.5. Ordne Größe
3.6. Ordne Art

4.0. Extras
4.1. Floppy anmelden: CONFIG ∗
4.2. Anwendung anmelden: CONFIG ∗
4.3. Voreinstellung: CONFIG ∗
4.4. Arbeit sichern: CONFIG ∗
4.5. Hardcopy ∗

Die Befehle COPY, ERASE und RMDIR werden nicht über die Menüleiste, sondern über die Desktop-Ikonen „Diskstationen" und „Papierkorb" ausgeführt. Es werden nämlich zusätzlich zur Menüleiste nach dem Booten der Systemdiskette mindestens 2 Laufwerk-Ikonen („Diskstation A" usw.) sowie die Lösch-Ikone („Papierkorb") angezeigt.
Die meisten Operationen lassen sich nun durch Zeigen mit dem Mauspfeil und einfaches Klicken auf Ikone und Menü-Option oder bei bestimmten Befehlen (DIR, RUN) alternativ durch doppeltes Klicken auf Ikone auslösen. Den Mauspfeil kann man auch ohne Maus mit den Cursortasten in Verbindung mit anderen Tasten (Alternate, Shift usw.) bewegen. Ferner kann man auch das Klicken durch Tastenkombinationen bewerkstelligen. Leider wurde diese Alternative nicht konsequent genug durchgeführt. Ein Beispiel hierfür ist die Fehlermeldung „Keine Disk in den Papierkorb", denn hier wurde vergessen, zusätzlich die Return-Taste abzufragen, wie dies bei den meisten anderen Fehlermeldungen der Fall ist (vgl. Handbuch, S. 31).

Der Komfort der GEM-Anwenderschnittstelle kann über die Tatsache nicht hinwegtäuschen, daß eine Reihe von wünschenswerten Befehlen fehlen, z.B. DISKCOMP, FILCOMP u.a., oder nicht angemessen implementiert worden sind, z.B. REN in bezug auf Volume- und Subdirectory-Namen, DISKCOPY mit FORMAT u.a. Außerdem muß man speziell bei größeren Datenträgern, also z.B. der 20M-Festplatte, bei manchen Befehlen starke Komforteinbußen hinnehmen. Zwei Beispiele seien hier herausgegriffen:
– Wenn man mit verschachtelten Directories arbeitet, muß man sich durch die einzelnen Directories klicken, bis man bei einem tiefen Subdirectory angelangt ist. Ähnlich ergeht es einem, wenn man zu einem höheren Directory zurückschalten will. Hier wären ein PREFIX- sowie ein TREE-Befehl wünschenswert.
– Da man immer nur das kopieren/löschen kann, was man als Ikone sieht, ist ein selektives Kopieren/Löschen einer Reihe von Dateien, deren Ikonen weit verstreut sind, dann nicht möglich, wenn sie in das maximal geöffnete Directory-Fenster nicht mehr alle hineinpassen. Wenn man nämlich beispielsweise hochscrollt, verschwinden die unten bereits markierten Ikonen aus dem Fenster und werden deshalb beim Kopieren/Löschen ignoriert.

2. Die Befehle

Nachfolgend werden die Datei- und Index-Optionen sowie einige weitere Befehle nach der MS-DOS-Nomenklatur beschrieben. Wegen der mit „∗" markierten Spezial-Optionen sei auf das Handbuch verwiesen.

FORMAT
Eine Diskette wird formatiert, indem die Diskstation-Ikone angeklickt und dann die Option 2.6 gewählt wird, worauf ein Warn- und Formatierfenster erscheint. Es können wahlweise einseitige 360K- oder beidseitige 720K-Disketten formatiert werden. Der vor dem Formatieren festgelegte Diskettenname (Label) kann nachträglich nicht mehr verändert werden. Obwohl das Formatieren eine Diskette löscht, kann man die zu formatierende Diskette nicht in den Papierkorb stecken, wie man annehmen müßte. Man muß also erst einmal *lernen*, was die Ikonenoperationen bedeuten. Intuition hilft hier wenig.

DISKCOPY
Eine Diskette wird dupliziert, indem das Disk-A-Bild (Quelldiskette) über das Disk-B-Bild (Zieldiskette) placiert wird oder umgekehrt, worauf ein Warn- und Kopierfenster erscheint. Die gleiche und damit recht irreführende Operation gilt auch für 1-Drive-Besitzer, die als Zieldiskette Laufwerk B (für von A nach B) oder A (für von B nach A) angeben müssen, obwohl das zweite Laufwerk nicht existiert. Die Konfusion rührt daher, daß mit A und B in diesem speziellen Fall nicht die Laufwerke, sondern die Disketten, die beide abwechselnd in Drive A gesteckt werden müssen, gemeint sind. Ferner werden die Disketten bei der 1-Drive-Kopie nicht überprüft, so daß man gegen Verwechslungen nicht geschützt ist und deshalb vorsichtshalber bei der Quelldiskette den Schreibschutzschieber nach oben drückt.
Die Diskettenkopie gehört nicht zu den in der Menüleiste aufgeführten Optionen. Ferner sei darauf hingewiesen, daß Formatieren und Kopieren getrennte Vorgänge sind.

DIR und CD
Der Directory-Befehl ist sehr komfortabel und variationsreich. Durch Anklicken der entsprechenden Disk-Ikone (z.B. Station A) und Wahl der Öffne-Option (2.1) wird das Volume-Directory (z.B. von Disk A) in einem Directory-Fenster angezeigt. Die Dateieinträge eines Inhaltsverzeichnisses (eigentliche Dateien sowie Subdirectories) können über die Index-Option (3.0) als Ikonen mit Namen (3.1) oder als Textzeilen mit Längenangabe und Datum (3.2) sowie innerhalb der Ikonen/Textzeilen nach Namen (3.3), Datum (3.4), Größe (3.5) oder Art (3.6) sortiert angezeigt werden. Es gibt unterschiedlich stilisierte Eintrag-Ikonen für Datendateien („Notizblock

mit Knick"), Programmdateien („Notizblock ohne Knick*") und Subdirectories („Ordner"). Im Gegensatz zum Macintosh lassen sich Eintrag-Ikonen nicht innerhalb des Directory-Fensters verrücken.

Neben Volume-Directories lassen sich auch Subdirectories öffnen (ebenfalls über Option 2.1), doch können nur maximal 4 Directory-Fenster offen sein. Mit dem Öffnen des Subsubdirectory-Fensters der Diskette A wird das nächsthöhere Subdirectory-Fenster automatisch geschlossen. Andernfalls wäre es nämlich nicht möglich, von einem Subsubsubdirectory-Fenster in ein anderes Subsubsubdirectory-Fenster zu kopieren, denn hier wie bei den anderen Desktop-Befehlen gilt der Grundsatz, daß sich Befehle nur auf sichtbare Objekte erstrecken können. So ist etwa „Blindkopieren" von nicht im Directory-Fenster sichtbaren Dateien unmöglich. Darin unterscheidet sich die GEM-Anwenderschnittstelle von der DOS-Programmiererschnittstelle.

Nicht mehr benötigte Directory-Fenster können über die Schließe- (2.4) und Schließe-Fenster-Optionen (2.5) wieder vom „Schreibtisch" (Desktop) entfernt werden. Als Skurrilität sei vermerkt, daß sich *dasselbe* Directory-Fenster bis zu viermal *gleichzeitig* öffnen läßt. Befehle wie Rename und Erase wirken sich dann tatsächlich auf alle vier gleichen (oder dieselben?) Fenster *gleichzeitig* aus. Dies war wohl ursprünglich nicht im Sinne des Erfinders! Da jedoch nur maximal 4 Fenster geöffnet sein können, *muß* man dasselbe Volume-Directory mehrfach öffnen können, wenn bei *derselben* Diskette von einem Subdirectory bestimmten Grades in ein anderes Subdirectory anderen Grades Dateien kopiert werden sollen.

VOL und REN

Die Zeige-Info-Option (2.2) dient
– einerseits zur Anzeige der belegten Bytes eines ganzen Volume-Directory (= einer ganzen Diskette), eines ganzen Subdirectory (= Summe aller Subdirectory-Files) oder einer einzelnen Datei, sowie
– andererseits zum Ändern eines Dateinamens.

Dagegen können Volume- und Subdirectory-Namen zwar angezeigt, aber nachträglich nicht mehr geändert werden, es sei denn durch Formatieren und Umkopieren auf eine Zweitdiskette (vgl. hierzu die Falschangabe im Handbuch, S. 42). Dies ist lästig, denn schließlich kann sich jeder einmal bei der Vergabe eines Subdirectory-Namens vertippen.

Das Verfahren der Namensänderung ist übrigens beim Macintosh anders, denn dort braucht man nur auf den Namen zu klicken, um ihn dann direkt abändern zu können.

MKDIR

In dem jeweils aktiven Volume- oder Subdirectory-Fenster läßt sich ein Ordner beliebigen Namens über die Neuer-Ordner-Option (2.3) anlegen. Es handelt sich dabei um echte Subdirectories wie beim MS-DOS-Betriebssystem sowie beim neuen Finder des Macintosh Plus (im Gegensatz zu den Pseudo-Subdirectories beim alten Macintosh-Finder).

ERASE und RMDIR

Um eine Datei oder ein Directory zu löschen, muß man die Datei- oder Ordner-Ikone über die Papierkorb-Ikone placieren. Man kann einzelne Dateien, Dateigruppen, ganze Directories (also nicht nur leere Directories) sowie sogar Directory-Gruppen auf einen Schlag löschen – nach entsprechender Vorwarnung, versteht sich!
Der ERASE-Befehl ist nicht aus der Menüleiste ersichtlich.

COPY

Um eine Datei in ein anderes Directory zu kopieren, muß man die entsprechende Ikone aus dem Quell-Directory in das geöffnete Ziel-Directory oder auf die Ziel-Ordner-Ikone placieren. Wenn man ein ganzes Subdirectory in ein Ziel-Directory kopiert, so wird das noch nicht vorhandene Ziel-Subdirectory automatisch erzeugt. Darüber hinaus kann man eine Datei oder – weniger sinnvoll – ein Subdirectory in *demselben* Directory duplizieren, wobei man einen neuen Datei- oder Subdirectory-Namen vorgeben muß.

RUN und TYPE

Programmdateien können über die Öffne-Option (2.1) direkt gestartet werden (analog zum BASIC-RUN-Befehl). Falls es sich um Textfiles handelt, so können diese über den Bildschirm oder auch über den Drucker ausgegeben werden (TYPE-Befehl).

II. DOS-Shell

Als Beispiel für eine MS-DOS-ähnliche Command-Line-Schnittstelle für Programmierer stellen wir nachfolgend das DOS-Shell-Programm von Timothy Purves vor, das in diesem Jahr bei der Firma Michtron in den USA erschienen ist.

1. Allgemeines

Zunächst einige vergleichende Hinweise für alle, die schon flüchtig von den anderen Betriebssystemen MS-DOS, CP/M, DOS 3.3, ProDOS und UCSD gehört haben. Eine detailliertere Analyse des Atari-Betriebssystems folgt zu einem späteren Zeitpunkt.

Am Atari ST können 2 Laufwerke und 1 Festplatte angeschlossen sowie im RAM eine virtuelle Diskette (RAM-Disk) angelegt werden. Diese Datenspeicher werden wie unter CP/M mit den Laufwerkkennungen A, B, C, D usw. angesprochen. So gilt für das eingebaute 1. Laufwerk die Kennung A und für das externe 2. Laufwerk die Kennung B. Diese Laufwerkkennungen entsprechen den Slot-Drive-Angaben unter DOS 3.3 und ProDOS (S6, D1 usw.) sowie UCSD (#4, #5 usw.). So wird etwa mit
DIR A:
das Inhaltsverzeichnis derjenigen Diskette angesprochen, die sich *zufällig* im Laufwerk A befindet. Die Kennung A markiert also nicht die Diskette selbst, denn sie könnte man ja auch in das Laufwerk B eingelegt haben. Aus diesem Grunde kann man beim Atari ST, wenn nicht gerade Dateien gelesen werden oder noch offen sind, in der Regel unbedenklich Disketten in demselben Laufwerk austauschen. Dies ist unter anderen Betriebssystem (ProDOS, CP/M) nicht ohne weiteres möglich. (Das Atari-Betriebssystem „merkt" jedoch einen Diskettenwechsel an der Diskettenerkennungsnummer.)
Eine Diskette kann zwar unter MS-DOS und Atari-DOS einen Namen haben, der als *Label* bezeichnet und bei der Formatierung festgelegt wird, z.B. DISK007. Mit diesem Namen kann eine Diskette jedoch nicht angesprochen werden, da das Label gewissermaßen nur als aufgeklebtes Etikett fungiert. Es hat damit nicht dieselbe Aufgabe wie der Volume-Name unter ProDOS. Folglich zeigt der Befehl
DIR A:
zwar das Hauptinhaltsverzeichnis (Volume-Directory) der Diskette im Laufwerk A an, doch wäre der analoge ProDOS-Befehl
CAT DISK007
ungültig. Die A/B/C- und Label-Konvention sind damit noch Relikte aus älteren Betriebssystemen. Ferner fällt auf, daß weder das Label noch Subdirectories ohne Tricks umbenannt werden können.

Der wesentliche Unterschied zwischen Apple-ProDOS und Atari-TOS ist folgender:

– ProDOS als lupenreines hierarchisches Betriebssystem findet vollautomatisch eine durch den vollständigen Pfadnamen

* soll Desktop symbolisieren

(Vol/Dir/Subdir/File) bestimmte Datei auf jedem angeschlossenen Laufwerk, gleichgültig in welchem Laufwerk sich die Diskette befindet, weil über den beim Systemstart ausgeführten Online-Befehl die Volume-Namen aller Datenträger, d.h. aller eingelegten Disketten, registriert werden. ProDOS-Befehle sind damit völlig laufwerkunabhängig.

– TOS als fast reines hierarchisches Betriebssystem findet eine bestimmte Datei nur dann, wenn zusätzlich zum Pfadnamen das Laufwerk spezifiziert wird (A:\Dir\Subdir\File). TOS-Befehle sind damit laufwerkabhängig. Folgerichtig sind getrennte Kommandos für Directory-Wechsel (CHDIR) und Drive-Wechsel (CHDRIVE) implementiert.

Das Hauptinhaltsverzeichnis (Volume-Directory, Root-Directory; Kennung A usw.) einer Atari-Diskette kann wie unter ProDOS Unterinhaltsverzeichnisse (Subdirectories) enthalten, die ihrerseits wiederum Unterinhaltsverzeichnisse (Subsubdirectories) enthalten können (hierarchische Directory-Struktur). Dabei entspricht der ProDOS-Notation
/DISK007/SUBDIR/SUBSUBDIR
die Atari-Notation
A:\SUBDIR\SUBSUBDR
D.h. für das Hauptinhaltsverzeichnis tritt die jeweilige Laufwerkkennung ein, und anstelle von „/" wird „\" verwendet.

2. Die Befehle

Nach dem Start meldet sich das DOS-Shell-Programm mit dem Ersatzlaufwerk und „>" als Bereitschaftszeichen (Prompt) sowie mit einer blinkenden Schreibmarke (Cursor), also z.B. so
A>□
Danach kann man jeweils einen Befehl, gefolgt von eventuellen Parametern, eingeben:

Systembefehle:
HELP, EXIT
CLS, VER
VERIFY
PROMPT
DATE, TIME

Directory-Befehle:
VOL
CHKDSK CK
TREE, DIR
CHDIR CD
MKDIR MD
RMDIR RD

Datei-Befehle:
COPY
ERASE DEL
RENAME REN
TYPE

Stapelbefehle:
BUILD
REM
ECHO, PAUSE
GOTO, IF, SET, SHIFT
FIND, MORE, SORT
RUN, PATH

Die Befehle, von denen wir nachfolgend die wichtigsten beschreiben werden, lassen sich formal in solche mit Parametern (m.P.) und solche ohne Parameter (o.P.) einteilen.

2.1. Systembefehle

A:
Das Umschalten von Laufwerk B oder C nach A usw. geschieht durch Eingabe des entsprechenden Buchstabens, gefolgt von „:". Das spezifizierte Laufwerk wird dann zum Ersatzlaufwerk (Default-Drive), so daß beispielsweise bei DIR die Laufwerkangabe entfallen kann. Beispiele:
B:
DIR
stellt auf B um und zeigt danach Inhaltsverzeichnis von B an.
A:
schaltet auf A zurück.

HELP o.P.
gibt eine alphabetische Befehlsliste aus. Man tippt hier wie bei allen anderen parameterlosen Befehlen lediglich das nackte Befehlswort ein.
EXIT o.P.
dient zum Verlassen des DOS-Shell-Programms.

CLS o.P.
löscht den Bildschirm.
VER o.P.
zeigt die TOS-Version an.

VERIFY (ON oder OFF)
bezieht sich auf den Verify-Status. Unter Verify versteht man das nochmalige, überprüfende Lesen/Schreiben beim Kopieren von Dateien usw.
VERIFY
zeigt den Status (on/off) an.
VERIFY OFF
schaltet Verify ab. Danach werden Dateien spürbar schneller kopiert.
VERIFY ON
schaltet Verify wieder ein.

PROMPT m.P.
ändert das Bereitschaftszeichen, normalerweise A>, B> oder C>, je nach aktivem Laufwerk. Die Syntax ist kompliziert. Mit z.B.
PROMPT $p
wird neben A> zusätzlich das aktuelle Subdirectory angezeigt. Mit
PROMPT >
gelangt man zum alten Prompt A> zurück. Näheres entnehme man der DOS-Shell-Anleitung oder entsprechenden MS-DOS-Büchern (s.u.)

DATE o.P
zeigt das Datum an. Danach erscheint eine Eingabeanweisung mit dem Format des einzugebenden Datums. Gibt man nur Return ein, so bleibt das alte Datum erhalten. Der Befehl
TIME o.P
funktioniert entsprechend.

2.2. Directory-Befehle

VOL (Drive-Nr.)
zeigt den Diskettennamen an, der beim Formatieren festgelegt worden ist, z.B.
VOL B:
Die Anzeige hat die Form
Volume in Drive B: is DISKNAME.
Der LABEL-Befehl, mit dem man unter MS-DOS den Diskettennamen nachträglich ändern kann, fehlt beim DOS-Shell.

CHKDSK (Drive-Nr.)
CK (Drive-Nr.)
überprüft das Inhaltsverzeichnis einer Diskette auf Integrität und zeigt danach den Diskettennamen, die Anzahl der freien und belegten Bytes sowie alle Unterinhaltsverzeichnisse (TREE-Befehl, s.u.) an. Beispiele:
CHKDSK B:
bezieht sich auf das Laufwerk B.
CK
bezieht sich auf das Ersatzlaufwerk, z.B. A.

TREE (Drive-Nr.)
zeigt die Baumstruktur des Inhaltsverzeichnisses an, z.B.
TREE B:
Der TREE-Befehl ist bereits im CHKDSK-Befehl enthalten.

DIR (m.P.)
zeigt das Inhaltsverzeichnis an, d.h. die Dateinamen mit Dateilänge, Datum usw. Beispiele:
DIR
zeigt das momentane bzw. durch CD (s.u.) festgelegte Inhaltsverzeichnis an.
DIR B:\
zeigt das Hauptinhaltsverzeichnis (Volume-Directory) der Diskette in Laufwerk B an.
DIR B:
zeigt nur dann das Hauptinhaltsverzeichnis an, wenn nicht zuvor mit CD ein Unter-

inhaltsverzeichnis (Subdirectory) spezifiziert worden ist.
DIR A:\SUBDIR\SUBSUBDR
zeigt das im Subdirectory SUBDIR enthaltene Subsubdirectory SUBSUBDR der Diskette A an.

∗.∗ und ????????.???
Mit Stellvertreterzeichen (Wildcards) können beim DIR-Befehl auch Dateinamen gefiltert bzw. selektiert werden, wie dies bereits unter CP/M üblich war. Der vollständige Dateiname, z.B.
DATEINAME.BAS
gliedert sich in den Dateinamen im engeren Sinne (max. 8 Zeichen) und die Dateitypkennung (File-Extension, max. 3 Zeichen). Ein Punkt trennt beide Bestandteile (nur bei der Befehlseingabe, nicht physikalisch auf der Diskette). Die Filterung geschieht dann mittels „∗" und „?".
Einige „∗"-Beispiele:
DIR ∗.∗
zeigt *alle* Dateien an (entspricht DIR o.P.).
DIR ∗.BAS
zeigt nur die Dateien mit der Dateikennung BAS an.
DIR Z∗.∗
zeigt nur die Dateien mit dem Anfangsbuchstaben „Z" an.
DIR A∗.B∗
zeigt nur diejenigen Dateien an, deren Dateiname mit „A" und deren Kennung mit „B" beginnen.
Aus den Beispielen lernen wir, daß „∗" für eine Zeichen*folge* steht. Demgegenüber tritt „?" für ein *einzelnes* Zeichen ein.
„?"-Beispiele:
DIR ????????.???
zeigt *alle* Dateien an (entspricht DIR o.P.).
DIR ∗.??S
zeigt nur diejenigen Dateien an, deren dritter Kennbuchstabe „S" ist.

CHDIR Directory-Name
CD Directory-Name
spezifizieren das Ersatz-Inhaltsverzeichnis (Default-Prefix). Beispiele:
CD \
geht zurück zum Hauptinhaltsverzeichnis (Volume-Directory) bzw. Wurzelverzeichnis (Root-Directory).
CD \BASIC
stellt auf das Unterinhaltsverzeichnis (Subdirectory) BASIC ein.
CD \BASIC\DEMOS
stellt auf das Subsubdirectory DEMOS innerhalb des Subdirectory BASIC ein.
CD ..
also CD mit 2 Punkten, geht zum nächsthöheren Subdirectory zurück, d.h. z.B. von \BASIC\DEMOS zu \BASIC.
CD o.P.
zeigt das momentane Directory an, z.B. in der Form
A:\BASIC

Hinweis: Das Umschalten von Laufwerk A nach Laufwerk B geschieht, wie oben bereits erwähnt, durch Eingabe von
B:
Wenn nun aber das Ersatzlaufwerk A war und man z.B.
CD B:\BASIC
eingibt, so bezieht sich der nächste DIR-Befehl noch *nicht* auf das Subdirectory BASIC von Laufwerk B, sondern weiterhin auf Laufwerk A. Gibt man jetzt jedoch
B:
und danach
DIR
ein, so bezieht sich DIR nunmehr auf das Subdirectory BASIC in Laufwerk B.

MKDIR Subdirectory-Name
MD Subdirectory-Name
erzeugen ein neues Unterinhaltsverzeichnis. Beispiele:
MD B:\SUBDIR
MD B:\SUBDIR\SUBSUBD
Die Laufwerkskennung B kann hier auch weggelassen werden, wenn B bereits Ersatzlaufwerk ist.
RMDIR Subdirectory-Name
RD Subdirectory-Name
entfernen ein altes Inhaltsverzeichnis, das jedoch keine Dateien oder weitere Subsubdirectories enthalten darf. Beispiele:
RD B:\SUBDIR\SUBSUBD
RD B:\SUBDIR

2.3. Dateibefehle

COPY Quelle Ziel
dient zum Kopieren von Dateien. Für die Quelldatei(en) gelten die oben erläuterten „∗"- und „?"-Dateinamen-Selektionsmethoden. Eine Datei kann zwar nicht auf sich selbst kopiert werden, doch kann eine Datei in dasselbe Directory unter Namensänderung dupliziert werden. Beispiele:
COPY XXX.BAS YYY.BAS
dupliziert Datei XXX.BAS im momentanen Subdirectory zur Datei YYY.BAS.
COPY A:\SUB\XXX.BAS B:YYY.BAS
kopiert Datei XXX.BAS aus Subdirectory SUB der Diskette A in das Volume-Directory der Diskette B unter dem neuen Namen YYY.BAS
COPY A:\SUBA\∗.BAS B:\SUBB
kopiert alle Dateien mit der Kennung BAS aus dem Subdirectory SUBA der Diskette A in das Subdirectory SUBB der Diskette B. Man beachte, daß auf SUBB kein Stellvertreterzeichen folgt.

ERASE Dateiname
DEL Dateiname
Mit diesem Befehl können Dateien (nicht Directories!) gelöscht werden. Beispiele:
ERASE B:\BASIC\DEMO.BAS
DEL XXX.BAS
DEL B:\BASIC\∗.BAS
Man beachte, daß beim ERASE- sowie auch beim COPY-Befehl die Dateikennung stets angegeben werden muß. Wenn die Stellvertreterzeichen „∗" oder „?" verwendet werden, so wird sicherheitshalber nach Y(es), N(o), G(lobal) oder Q(uit) gefragt.

RENAME Altname Neuname
REN Altname Neuname
Mit diesen Befehlen kann man Dateinamen, nicht jedoch Volume- und Subdirectory-Namen, umbenennen. Beispiele:
RENAME XXX.BAS YYY.BAS
ändert den Dateinamen XXX.BAS in YYY.BAS um.
RENAME DEMO.BAS DEMO.PRG
RENAME DEMO.BAS DEMO1.PRG
Die Wildcards „∗" und „?" sind nicht zulässig, also *nicht*
RENAME B:∗.BAS B:∗.BIS

TYPE Dateiname
Damit können Textdateien, also Files im ASCII-Format, auf den Bildschirm ausgegeben werden. Mit Ctrl-S kann die Anzeige angehalten, mit Ctrl-Q fortgesetzt und mit Ctrl-C abgebrochen werden. Gibt man einen Nicht-Textfile vor, so wird nur „Schrott" angezeigt. Leider kann man an der Dateitypkennung nicht zuverlässig erkennen, ob ein Textfile vorliegt, zumal man die Kennung (mißbräuchlich) umbenennen kann. Beispiel:
TYPE DESKTOP.INF
Dies ist zufällig ein normaler Textfile.

2.4. Stapelbefehle

BUILD Name.BAT

Mit diesem Befehl kann man Befehlsdateien anlegen, deren Dateikennung „BAT" heißen muß. Dies geschieht wie folgt:
BUILD BEFEHLE.BAT
Danach wird „1:" ausgegeben, und man kann den 1. String eingeben. Daraufhin wird „2:" ausgegeben, und man kann den 2. String eingeben, usw. Beispiel:
1: DIR
2: TREE
3: Ctrl-C
Die letzte Eingabe wird mit Ctrl-C abgebrochen. Danach befindet sich die Datei namens BEFEHL.BAT auf der Diskette. Nunmehr kann man durch Eingabe des reinen Befehlsdateinamens
BEFEHL.BAT
(nicht mit RUN!) die Befehlsdatei automatisch abarbeiten lassen (Stapelverfahren, Batch-Processing). Befehlsdateien entsprechen den Execute-Dateien in anderen Betriebssystemen (DOS 3.3, ProDOS, UCSD). Eine BAT-Datei kann beispiels-

weise eine Reihe von COPY-Befehlen enthalten, so daß man beim Programmstart die gewünschten Dateien in die RAM-Disk überspielen kann.
Da der BATCH-Befehl nur über begrenzte Editiermöglichkeiten verfügt, kann man die Befehlsdateien auch mit einem Textverarbeitungsprogramm erstellen, das ASCII-Textfiles erzeugt. Bei „1st Word" muß man zu diesem Zweck den Word-Processing-Modus abstellen.
Aus Platzgründen müssen wir an dieser Stelle auf die Beschreibung der anderen Batch-Befehle verzichten.

Sybex-MS-DOS-Ratgeber

Im Sybex-Verlag ist in der Ratgeber-Reihe ein handliches Nachschlagewerk unter dem Titel „MS-DOS" von Wolfgang Höfs (1986, 455 S., DM 55,-) erschienen, das in sehr übersichtlicher und gelungener Form alle MS-DOS-Befehle (sowie darüber hinaus im zweiten Teil die Systemaufrufe) beschreibt. Die obige Darstellung des DOS-Shell-Programms wurde mit dem Sybex-Buch abgeglichen. Danach fehlen viele wichtige Diskettenbefehle wie FORMAT, DISKCOMP, DISKCOPY, FILCOMP usw.

Programme in der Art von DOS-Shell sind nützliche Hilfen, insbesondere für Programmierer, die die Quick-and-Dirty-Methode der Maus- und Ikonenmethode vorziehen. Doch müßten weitergehende Befehle implementiert sein, um von einer rundum befriedigenden Programmiererschnittstelle reden zu können. Als wesentlicher Vorzug gegenüber der GEM-Anwenderschnittstelle ist die Abarbeitung von Batch-Dateien zu nennen, mit deren Hilfe man gewisse, regelmäßig vorkommende Disketten- und Dateibearbeitungen automatisieren kann.

Wir suchen noch

Atari-Autoren

Wer fundierte und gut gegliederte Fachaufsätze schreiben kann, melde sich bitte bei der Peeker-Redaktion.

06221 / 489352

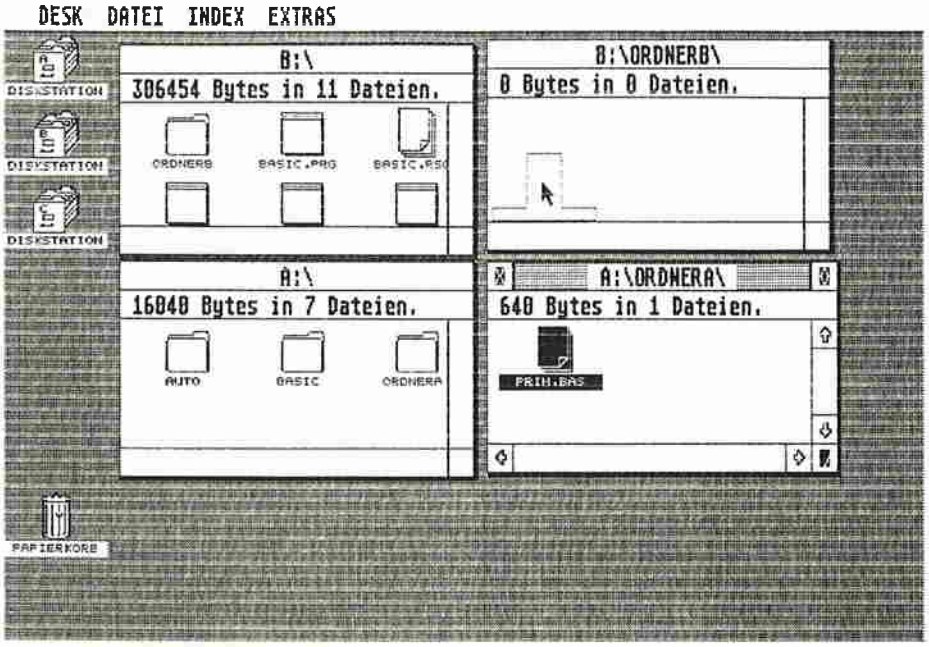


*Bild 1: Kopie von Datei PRIM.BAS **aus geöffnetem** ORDNER A der Diskette in Drive A **in geöffneten** ORDNER B der Diskette in Drive B.*

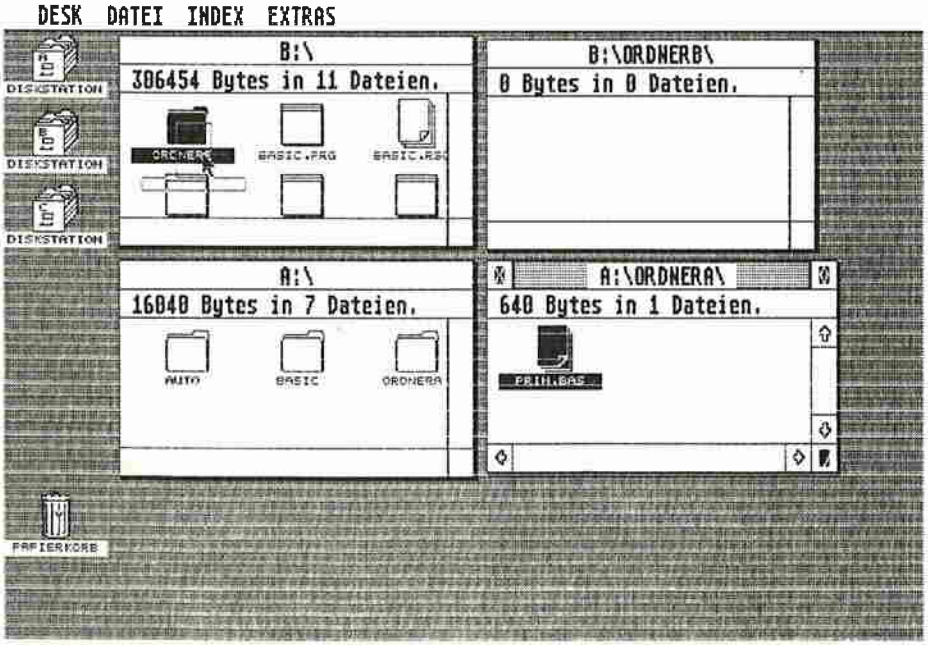


*Bild 2: Kopie von Datei PRIM.BAS **aus geöffnetem** ORDNER A der Diskette in Drive A **auf geschlossenen** ORDNER B der Diskette in Drive B. Der Zielordner darf also hier im Gegensatz zum Quellordner geschlossen sein.*

Atari-Händler

Die mit * gekennzeichneten Atari-Händler sind zugleich Apple-Händler; ehemalige Apple-Händler sind mit ** gekennzeichnet.

Nach Postleitzahlen sortiert.

Alpha Computers, E. Muth GmbH, 1000 Berlin 31, Kurfürstendamm 121a
Computercenter Karstadt AG, 1000 Berlin 61, Hermannplatz**
Computare Elektronik OHG, 1000 Berlin 30, Keithstraße 18-20
Runow Büroelektronik, 1000 Berlin 30, Keithstraße 26*
Union-Zeiss, 1000 Berlin 15, Am Kurfürstendamm 57
Bit Computershop GmbH, 2000 Hamburg 20, Osterstraße 173**
Computercenter Karstadt AG, 2000 Hamburg, Mönckebergstraße 16
Createam Microcomputer, 2000 Hamburg 71, Bramfelder Chaussee 300
G.M.A. mbH, 2000 Hamburg 76, Wandsbeker Chaussee 58
Haba Technik, 2000 Hamburg 54, Münsterstraße 9
Sellhorn GmbH, 2000 Norderstedt, Ulzburger Straße 2**
BPO Ges. f. Planung + Org. GmbH, 2080 Pinneberg, Dingstätte 34
Sienknecht Untern. Ber. f. Büro., 2120 Lüneburg, Heiligengeiststraße 20
Ernst Bergau GmbH, 2160 Stade, Neue Straße 5
Der Computerladen, 2210 Itzehoe, Holzkamp 12
MCC Micro Computer Christ, 2300 Kiel 1, Dreiecksplatz 7
RCM Computer Magazin, 2350 Neumünster, Klosterstraße 2
FK Bürotechnik, 2370 Rendsburg, Thormannplatz 20/22
Elektronic-Computer-Laden OHG, 2390 Flensburg, Norderstraße 94-96
Jessen und Lenz, 2400 Lübeck, Wahmstraße 36*
Brinkmann, 2800 Bremen, Obernstraße 82/88
PS-Data, 2800 Bremen, Doventorsteinweg 41
W. Walder & W.Manske Comp. Bed, 2849 Lutten, Auenstraße 4
Kurt Neumann Personalcomputer, 2850 Bremerhaven, Georgstraße 71
Carl Woeltje GmbH & Co. KG, 2900 Oldenburg, Heiligengeiststraße 6
Radio Tiemann GmbH & Co.KG, 2940 Wilhelmshaven, Marktstraße 52
S&F Datentechnik GmbH & Co.KG, 2950 Leer, Augustenstraße 3
Com Data, 3000 Hannover, Am Schiffergraben 19
Trenddata Computer GmbH, 3000 Hannover 1, Am Marstall 18-22
F&T Computervertrieb, 3040 Soltau, Am Hornberg 1
Büro Zentrum Haupt, 3100 Celle, Zöllnerstraße 2
Büro Zentrum Haupt, 3100 Celle, Gerhard-Kamm-Straße 2
W.S.G. Bürosysteme, 3150 Peine, Hagenstraße 34
Computer-Haus Gifhorn, 3170 Gifhorn, Braunschweiger Straße 50
Computercenter Horten AG, 3200 Hildesheim, Almsstraße 41
Witte Bürotechnik, 3250 Hameln, Bäkkerstraße 22
Computer Studio GmbH, 3300 Braunschweig, Rebenring 49-50*
Fricke Ber. u. Vertr. f. Computer, 3320 Salzgitter, Berliner Straße 54
Dr. Hildebrandt & Buchholz, 3380 Goslar, Rosentorstraße 31
Büroeinr.-Zentrum Widerholdt, 3400 Göttingen, Wagenstieg 14
Computercenter Kaufhof AG, 3500 Kassel, Obere Königsstraße 31**
Hermann Fischer OHG, 3500 Kassel, Rudolf-Schwander-Str. 5*
LWM Computer Service, 3550 Marburg/Lahn, Biegenstraße 43
Computercenter Horten AG, 4000 Düsseldorf, Berliner Allee 52**
Data Becker, 4000 Düsseldorf, Merowingerstraße 30*
E. Bernshaus GmbH + Co, 4000 Düsseldorf 13, Cäcilienstraße 2
Hoco EDV Anlagen GmbH, 4000 Düsseldorf, Flügelstraße 47*
Computer Commerce, 4050 Mönchengladbach, Hindenburgstraße 249**
Computercenter Horten AG, 4100 Duisburg, Düsseldorfer Straße 32
Compi Datensysteme, 4100 Duisburg 11, Rathausstraße 10
Computercenter Horten AG, 4150 Krefeld, Ostwall 170/180
SVI-GmbH, 4150 Krefeld, Carl-Wilhelm-Straße 27
Computer Syst. Feldmann + Luft, 4190 Kleve, Emmericher Straße 223
Kamp Bürosysteme GmbH, 4200 Oberhausen 12 (Osterfeld), Vestische Straße 89/91*
Compi Datensysteme, 4220 Dinslaken, Bahnstraße 20-26
HI-Tronic, 4280 Borken, Neutor 3
Computercenter Karstadt AG, 4300 Essen, Limbecker Platz
Helmut Rennen GmbH & Co.KG, 4300 Essen, Altendorfer Straße 40
Computer Centrale, 4350 Recklinghausen, Douaistraße 1
Micro-Data Hintze, 4350 Recklinghausen, Herner Straße 24
Basis Computer Systeme GmbH, 4400 Münster, Daimlerweg 39
Computercenter Horten AG, 4400 Münster, Ludgeristraße 1
Bense KG, 4420 Coesfeld, Dieselstraße 10-12*
OCB Org.-U. Computerber. GmbH, 4422 Ahaus, Wallstraße 3
Computercenter Horten AG, 4500 Osnabrück, Wittekindstraße 23
Heinicke-Electronik, 4500 Osnabrück, Kommenderiestraße 120
Büro Studio Bolz, 4600 Dortmund, Brauhausstraße 4
CC Computerstudio GmbH, 4600 Dortmund 1, Elisabethstraße 5
City Elektronik, 4600 Dortmund, Güntherstraße 75
Computercenter Karstadt AG, 4600 Dortmund, Westenhellweg 30-36**
BO-Data Computer, 4630 Bochum, Querenburger Höhe 209
Beate Vollrath Computer, 4650 Gelsenkirchen, Kirchstraße 17+18
Büro Rüter, 4700 Hamm, Gustav-Heinemann-Straße 19-21**
GET Ges. für Elektronische TE, 4790 Paderborn, Im Schildern 15
CSF Computer & Software GmbH, 4800 Bielefeld 1, Heeperstraße 106-108
Microtec Ges.f.MC Vertr. mbH, 4800 Bielefeld 14, Paul-Schwarze-Straße 5
Computer Store, 4830 Gütersloh, Schulstraße 9
Frevert Computer-Systeme, 4920 Lemgo, Braker Mitte 9
Büromaschinen Braun am Rudolfplatz GmbH, 5000 Köln, Richard-Wagner-Str. 39
Computercenter Kaufhof AG, 5000 Köln, Hohe Straße 41-53*
Computerstudio Hölscher, 5010 Bergheim, Zeppelinstraße 7
Computer Center, 5060 Bergisch-Gladbach, Buchholzstraße 1
Rolf Rocke Computer, 5090 Leverkusen, Auestraße 1
Runte Bürotechnik, 5090 Leverkusen, Konrad-Adenauer-Platz 2
Allo Pach GmbH & Co.KG, 5100 Aachen, Adalbertstraße 82 + 92*
Computercenter Kaufhof AG, 5100 Aachen, Adalbertstraße 20-30**
Computer Center, 5200 Siegburg, Luisenstraße 26
Byte ME Computer Systeme, 5240 Betzdorf/Sieg, Wilhelmstraße 25
Computercenter Kaufhof AG, 5300 Bonn, Remigiusstraße 20-24*
H&G Hansen & Gieraths EDV GmbH, 5300 Bonn, Münsterstraße 1
Plasman Computercenter, 5300 Bonn 1, Heerstraße 175-179*
Texteam Computer und Textsyst. Zub., 5350 Euskirchen, Kessenicher Straße 43
Schmitt Computer Systeme, 5400 Koblenz, Casinostraße 40
Staadt Computertechnik, 5450 Neuwied, Mittelstraße 82
Dr. Aumann GmbH, 5451 Straßenhaus, Schulstraße 12
Bürocenter Lehr GmbH, 5500 Trier, Güterstraße 82*
Novocomp Datensyteme GmbH, 5500 Trier, Walramsnerstraße 7 + 9*
ATC Computersysteme, 5540 Prüm, Kalvarienbergstraße 34
Jung am Wall, 5600 Wuppertal 1, Wall 31-33
Meiers Büroorg. GmbH & Co.KG, 5600 Wuppertal, Friedrich-Engels-Allee 352
COM Soft, 5630 Remscheid, Schneiderstraße 12
Data Kamp GmbH, 5650 Solingen, Werwolf 4
Comp. Techn. Erkelenz u.Klug, 5800 Hagen 1, Hochstraße 96*
Hampf Electronic GmbH, 5880 Lüdenscheid, Hochstraße 1
Hees Computer Vertriebs GmbH, 5900 Siegen, Weidenauer Straße 72
BCO Büro-Computer + Org. GmbH, 6000 Frankfurt/Main 1, Oederweg 7-9
Computercenter Kaufhof AG, 6000 Frankfurt/Main, Zeil 116-126**
GES-Computer, 6000 Frankfurt/Main, Berger Str. 18
Müller & Nemecek GmbH, 6000 Frankfurt/Main, Kaiserstraße 44
Schmitt Computer Systeme, 6000 Frankfurt/Main, Große Friedberger Straße 30
Waizenegger GmbH & Co.KG, 6000 Frankfurt/Main, Kaiserstraße 41
Büro Emig GmbH, 6090 Rüsselsheim, Haßlocher Straße 26*
Computercenter Karstadt AG, 6100 Darmstadt, Elisabethenstraße 15**
Heim OHG-Bürotechnik, 6100 Darmstadt-Eberstadt, Heidelberger Landstraße 194
Schmitt Computer Systeme, 6100 Darmstadt, Mühlstraße 76
Computerhaus, 6143 Lorsch, Kastanienallee 10
Computercenter Karstadt AG, 6200 Wiesbaden, Kirchgasse 35-43
Schmitt Computer Systeme, 6200 Wiesbaden, Rheinstraße 41
KFC Computersysteme, 6240 Königstein, Wiesenstraße 18*
Computercenter Karstadt AG, 6300 Gießen, Seltersweg 64
Ernst Baums GmbH & Co.KG, 6300 Gießen, Bahnhofstraße 26
Weinrich Büroorg. GmbH & Co.KG, 6300 Gießen, Bahnhofstraße 26
KA-WE Computercenter, 6330 Wetzlar/Lahn, Bahnhofstraße 14
Computer Professional GmbH, 6350 Bad Nauheim, Hauptstraße 92
Computer Service Feil, 6390 Usingen, Obergasse 2*
Weinrich GmbH & Co., 6400 Fulda, Ronsbachstraße 22
Landolt-Computer, 6457 Maintal/Dörnigheim, Wingertstraße 112-114
Elphotec Computer Systeme GmbH, 6500 Mainz, Schießgartenstraße 7
Schmitt Computer Systeme, 6500 Mainz, Karmeliterplatz 4
Würth Büromarkt + Org. GmbH, 6500 Mainz, Rheinstraße 101-103*
Orion Computer Systeme, 6520 Worms, Friedrichstraße 22
Würth Bürobedarf + Org. GmbH, 6530 Bingen/Rhein, Am Rupertsberg 16*
Würth Bürobedarf + Org. GmbH, 6550 Bad Kreuznach, Schloßstraße 16*
Ringfoto Pullig, 6580 Idar-Oberstein, Hauptstraße 421*
Computercenter Karstadt AG, 6600 Saarbrücken, Bahnhofsstraße 82-100
W.N. Pfeiffer EDV-Abteilung, 6600 Saarbrücken 3, Beethovenstraße 28
W.N. Pfeiffer KG, 6640 Merzig 1, Schankstraße 30
W.N. Pfeiffer KG, 6650 Homburg, Saarbrücker Straße 7
W.N. Pfeiffer, 6660 Zweibrücken, Mühlstraße 4-6
W.N. Pfeiffer KG, 6690 St.Wedel, Mommstraße 7
MKV GmbH, 6700 Ludwigshafen, Im Bismarckzentrum*
MKV GmbH, 6720 Speyer, Gilgenstraße 4*
P.Felten & M.Meier Computersys., 6730 Neustadt/Weinstraße, Exterstraße 4
C.O.S. Computer Org. GmbH, 6750 Kaiserslautern, Karl-Marx-Straße 8
Gotthold Bürocenter, 6750 Kaiserslautern, Im Altstadtparkhaus
J-W Töns GmbH & Co.KG, 6780 Pirmasens, Hauptstraße 59
Computer Center, 6800 Mannheim, L14, 16-17**
Computercenter Horten AG, 6800 Mannheim, N7**
Gauch & Sturm, 6800 Mannheim, Casterfeldstraße 74-76
Heidelberger Comp. C. Micro-Data, 6900 Heidelberg, Bahnhofstraße 1
Jacom Comp. Techn. Vertr. GmbH, 6900 Heidelberg, Mönchhofstraße 3
BNT Computerf. Seibel & CO.OHG, 7000 Stuttgart 50, Marktstraße 48
Computercenter Kaufhof AG, 7000 Stuttgart, Königstraße 6 (Nähe Hbf)**
Kübler Büro- und Datentechnik, 7000 Stuttgart 1, Silberburgstr. 145
Matrai Computer, 7022 L.-Echterdingen, Bernhäuser Straße 8
MCA Computer Center, 7030 Böblingen, Sindelfinger Allee 1

Böhmer Electronic GmbH, 7080 Aalen, Wilhelm-Zapf-Str. 9*
Fritz Seel GmbH & Co.KG 7100 Heilbronn, Am Wollhaus 6
Walliser & Co, 7100 Heilbronn, Mönchseestraße 99
Weeske Computer-Electronic, 7150 Backnang, Potsdamer Ring 10
Büro-Org. Linde GmbH, 7170 Schwäbisch Hall, Heimbacher Gasse 5
Top Team EDV-Beratungsges. mbH, 7340 Geislingen/Steige, Hauptstraße 38
SRE-Ges. f. Datenverarbeit. mbH, 7450 Hechingen, Schloßplatz 3
Scheurer Heim + PC Comp.Markt, 7475 Meßstetten, Hauptstraße 10
Soft & Easy Computer GmbH, 7480 Sigmaringen, Rappgässle
MKV GmbH, 7500 Karlsruhe, Rüppurer Straße 2d*
Papierhaus Erhardt GmbH & Co.KG, 7500 Karlsruhe, Am Ludwigsplatz
Helmut Jöst HiFi-Video-Center, 7520 Bruchsal, Werner-von-Siemens-Straße 47a
DM-Technik Matekalo GmbH, 7530 Pforzheim, Durlacher Straße 39
HEWA Computercenter, 7537 Remchingen-Darmsbach, Feldbergstraße 34
Frank Leonhardt, Electronic, 7600 Offenburg, In der Jeuch 3, Gew.Geb.Walt.
Müntzer Elektronik, 7670 Neuried-Altenheim, Friedrichstraße 16
Udo Meier, 7700 Singen, Am Posthalterswäldle 8
Hartmut Schaber Comp.-Anlagen, 7707 Engen, Industriestraße 4a
Bus Brauch & Sauter Comp. Techn., 7730 VS-Schwenningen, Villinger Straße 85
Computertechnik Rösler, 7750 Konstanz, Zasiusstraße 35
CDS-EDV-Service GmbH, 7800 Freiburg, Windausstraße 2
Computer Aktuell, 7800 Freiburg, Kaiser-Joseph-Str. 232
Resin Büro mit System, 7851 Binzen, Am Dreispitz
Hettler-Data, 7890 Waldshut-Tiengen, Lenzburger Straße 4*
Computer Studio Claus Wecker, 7900 Ulm/Donau, Hafenbad 18/1
Hard and Soft Computer GmbH, 7900 Ulm/Donau, Herrenkellergasse 16
„das Büro" Bürosysteme GmbH, 7920 Heidenheim, Kastorstraße 11
Expert Stefan Doraszelski, 7920 Heidenheim, Wilhelmstraße 45/49
Computercenter Kaufhof AG, 8000 München Stachus, Karlsplatz 21**
Comp + Carry Comp-Syst. V. GmbH, 8000 München 45, Ingolstädter Straße 20-24*
Computercenter Kaufhof AG, 8000 München Marienplatz, Kaufinger Straße 1-5**
MSG Marketing und Service GmbH, 8000 München 82, Adelmannstraße 5
Ralf Ludwig Bürotechn. + Comp., 8000 München 45, Ingolstädter Straße 62L**
Schulz Computer, 8000 München 2, Schillerstraße 22
PRO-CE, 8032 Gräfeling, Pasinger Straße 94
Dreyer GmbH, 8070 Ingolstadt, Manchinger Straße 125
Radio Klement, 8120 Weilheim, Admiral-Hipper-Straße 1
Elektronik Center Bad Tölz, 8170 Bad Tölz, Wachterstraße 3
Computing, 8200 Rosenheim, Theodor-Gietl-Straße 3
Computer Studio G. Friedrich, 8220 Traunstein, Stadtplatz 10, Ludwigstraße 3*
Jaskulski, 8263 Burghausen, Mainstraße 243
Büro Dallmer, 8300 Landshut, Altstadt 69*
Elektroland Zimmermann, 8390 Passau, Meraner Straße 5
C-Soft GmbH, 8400 Regensburg, Holzfällerstraße 4*
Elektroland Zimmermann, 8400 Regensburg, Dr.-Gessler-Straße 8*
MCSC, 8480 Weiden i.d.Oberpf., Bahnhofstraße 43
A&P Shop, 8490 Cham, Auf der Schanze 4
Computercenter Karstadt AG, 8500 Nürnberg, Königstraße 14
Löw-Systems, 8508 Wendelstein, Äußere Fürther Straße 4a
Büro 2000 Haas, 8520 Erlangen, Dresdner Straße 5*
Haas im Zentrum, 8520 Erlangen, Friedrichstraße 9
Microcomputerservice, 8520 Erlangen, Meisenweg 29
Steinwald-Elektronik GmbH, 8590 Marktredwitz, Am Sterngrund 1
Bürozentrum A.+ R. Kutz, 8600 Bamberg, Am Kranen 12a
Computer-Center Burger, 8670 Hof, Leimitzer Straße 11
Haller GmbH, 8700 Würzburg, Büttnerstraße 29
Schöll Computercenter, 8700 Würzburg, Dominikanerplatz 3
Uhlenhuth GmbH, 8720 Schweinfurt, Albrecht-Dürer-Platz 2
Schöll Büroladen, 8730 Bad Kissingen, Münchner Landstraße 6
Victor Willgerodt GmbH & CO.KG, 8750 Aschaffenburg, Herrn V. Willgerodt
Computique, 8752 Mömbris, Im Kahltal-Zentrum
Radio-Busch GmbH, 8800 Ansbach, Uzstraße 8-10
ORG-Datentechnik GmbH, 8850 Donauwörth-RDL, Altes Sträßle 28
Adolf & Schmoll Computerstudio, 8900 Augsburg, Hörbrotstraße 6
Computercenter Karstadt AG, 8900 Augsburg, Bürgerm.- Fischer-Straße 6-10**
Video + Computer-Handels GmbH, 8901 Augsburg-Vogelsang, Steppacher Straße 8
Szeredy Elektronik GmbH, 8910 Landsberg am Lech, Sandauer Straße
EDV-Schweizer KG, 8940 Memmingen, Benninger Straße 34
Stählin Büro-Org., 8960 Kempten, Gerberstraße 32/34*
Büroma GmbH Hölzler + Schuler, 8990 Lindau, Rickenbacher Straße 7

Der nächste Peeker,
Heft 11/1986,
erscheint in der
letzten Oktoberwoche

Inserentenverzeichnis Peeker 10/86

aaa Electronic, Freiburg . . . . . Seite 47
Bühler Elektronik, Baden-Baden . . . . . Seite 33
Ecosoft Economy Software, Astano . . . . Seite 47
Frank & Britting GmbH, Forst . . . . . Seite 33
Klaus G. Gorny, Schweskau . . . . . Seite 33
IDW-Elecktronic, Neufahrn . . . . . Seite 33
Interkom Electronic, Isernhagen 4 . . . . Seite 33
Intus Lern-Systeme AG, Astano . . . . . Seite 47
Klaus Jeschke, Kelkheim . . . . . Seite 47
F. Mayer Computersysteme, Berlin . . . . Seite 71
MoVe GmbH, Leverkusen . . . . . Seite 33
MP//C-Datentechnik, Kerpen . . . . . Seite 47
Pandasoft, Berlin . . . . . Seite 61
M. Semjan Computer Systeme, Frankfurt . . Seite 57
Tewi-Verlag, München . . . . . Seite 11
Ueding Electronics, Menden . . . . . Seite 55
Weiss-Computer, Wilhelmshaven . . . . . Seite 46
Peeker-Börse . . . . . Seite 26

TurtleGraphics-Library-Paket

von Dieter Geiß

Turtle-Utilities für Fenstertechnik und Apple-Maus in einfacher und doppelter Hires-Grafik für Pascal 1.2 auf Apple IIe/c mit Maus oder Joystick. 2 Disketten mit umfangreichem Manual, DM 98,–. Unter Pascal 1.1 mit 64K nur eingeschränkt lauffähig

Im einzelnen bietet das Paket folgende Möglichkeiten:

- volle Kompatibilität mit der alten „TurtleGraphics"
- alle zeitkritischen Funktionen in reinem Assembler programmiert
- Benutzung der zweiten Hires-Seite ($4000-$5FFF) möglich
- für Apple IIc und Apple IIe mit erweiterter 80-Zeichen-Karte Benutzung der doppelten Hires-Grafik mit 560 × 192 Punkten bzw. 16 neuen Farben möglich
- schnelle Prozeduren zum Zeichnen eines Punktes oder einer Linie
- Benutzung mehrerer Zeichensätze gleichzeitig
- Scrolling des Hires-Schirms oder eines Teils in vier Richtungen
- drei verschiedene Schriftarten: Fett-, Breit- und Proportionalschrift, beliebig mischbar (acht Möglichkeiten)
- spezielle schnelle Ausgabe von Text
- Cursor bei Eingabe frei programmierbar
- Ein-/Ausgabe von INTEGER-, CHAR-, STRING- und REAL-Werten im Grafikmodus
- Menüzeile wie beim Macintosh
- Pull-down-Menüs
- Laden und Speichern von Fenstern (Windows)
- Öffnen von Fenstern
- Aktivieren und Deaktivieren von Fenstern
- Verschieben und Vergrößern/Verkleinern von Fenstern
- Scrolling von Fensterinhalten in allen vier Richtungen
- Umfangreiche Demos als Quelltexte.

Hüthig Software Service · Postfach 10 28 69 · 6900 Heidelberg

Sonderangebot für Peeker-Leser

Festplattenkomplettlösung für jedermann

Mit der Firma Frank & Britting GmbH, die auf Festplatten spezialisiert ist, konnten wir ein extrem günstiges Sonderangebot aushandeln, das eine Festplattenkomplettlösung selbst für Apple-Besitzer mit kleinerem Geldbeutel erschwinglich macht. Sie können unter zwei Varianten wählen:

Luxus-Lösung: 20-Megabyte-Festplatte MDB20 (MDB = Mobile Datenbox) + Megaboard-Controller + Handbuch + 3 Konfigurationsdisketten + Anschlußkabel + DB-Meister-Dateiverwaltungsprogramm + Handbuch + 2 Programmdisketten zum Gesamtpreis von nur DM 3199,- inkl. MwSt.

Standard-Lösung: 10-Megabyte-Festplatte MDB10 + Megaboard-Controller + Handbuch + 3 Konfigurationsdisketten + Anschlußkabel + DB-Meister-Dateiverwaltungsprogramm + Handbuch + 2 Programmdisketten zum Gesamtpreis von nur DM 2799,- inkl. MwSt.

(jeweils 6 Monate Garantie.)

Wozu eine Festplatte?

Für eine Festplatte sprechen zwei Argumente, nämlich erstens der bedeutend größere Datenspeicher und zweitens die sehr hohe Zugriffszeit.

● 80- und 160-Spur-Laufwerke haben zwar einen größeren Datenspeicher als die normalen 35-Spur-Laufwerke, doch ist die Zugriffszeit hier wie dort bescheiden.

● RAM-Karten haben zwar eine hohe Zugriffszeit, doch ist der Datenspeicher flüchtig: Strom aus – Daten weg!

Für unsere Festplatten-Standard-Lösung (MDB10) gilt demgegenüber:

– Die Nettospeicherkapazität beträgt hier ca. 9.650K = ca. 9.880.000 Zeichen = ca. 15 Disketten mit 160 Spuren = ca. 69 Disketten mit 35 Spuren.

– Die Datenübertragungsrate beträgt auf unterster Ebene ca. 44K/s = ca. 45.000 Zeichen/s. Zum Vergleich beläuft sich die Übertragungsrate bei normalen Diskettenlaufwerken auf ca. 7,5K/s und bei RAM-Karten auf ca. 44K/s, d.h. RAM-Karten sind also keineswegs schneller, sondern meist sogar langsamer als die MDB-Festplatte.

Sie erhalten deshalb mit unserer Festplattenlösung einen großen externen Massenspeicher bei sehr hoher Datenübertragungsrate zu einem äußerst niedrigen Preis.

Als Betriebssysteme können Sie im einzelnen verwenden:

– DOS 3.3,
– Diversi-DOS 2C und 4C,
– Apple-Pascal 1.1 und 1.2,
– ProDOS 1.0.1, 1.0.2, 1.1.1,
– CP/M 2.2 56K (z.Zt nur diese CP/M-Version).

DB-Meister

Speziell für die zukünftigen MDB-Besitzer wurde das Adreß- und Schemabriefprogramm DB-Meister, das in der normalen Diskettenversion DM 290,- kostet, zum Festplattenbetrieb umgeschrieben. Technische Daten:

– Datensatzlänge bis zu 230 Zeichen, aufteilbar in bis zu 25 Felder; maximal 1000 Datensätze pro Volume
– 2 variable Indexfelder und 1 zusätzliches, konstantes Suchfeld.
– Maskengenerator-Modul für die freie Definition von Bildschirmeingabe- und Ausdruck-Masken.
– Dateipflege-Modul zum Neueingeben, Ändern, Löschen, Ansehen, Bildschirm-Ausdruck usw.
– Sortier- und Filter-Module zum Sortieren und Untersortieren nach Indexfeldern und zum selektiven oder kumulierenden Filtern nach beliebigen Feldern.
– Druck-Modul zum Ausdrucken von Selbstklebe-Etiketten, tabellarischen Listen und Schemabriefen mit automatischem Einschub von Adressen und Anreden sowie weiteren Feldeinschüben im Briefkörper; Einzelblatt und Endlospapier verwendbar; Druckertreiber für Typenraddrucker mit anderer Typenradbelegung können eingebunden werden.
– Brief-Modul zum Schreiben der Schemabriefe; bei Bedarf separat verwendbar.
– Schnelles Backup-Programm (nur 10 Minuten für 2,5 Megabytes bzw. 18 DB-Meister-Dateien).

Festplatte läuft auch auf neuem Apple GS!

Wie wird bestellt?

Sie senden Ihre Bestellung an den Hüthig-Software-Service. Sie erhalten dann von der Firma Frank & Britting eine Vorausrechnung, nach deren Überweisung Ihnen von dort die MDB10 bzw. MDB20, der Megaboard-Controller, das Handbuch und die Konfigurierungsdisketten mit 6 Monaten Garantie geliefert werden. Gleichzeitig erhalten Sie vom Hüthig-Software-Service das DB-Meister-Programm (2 Disketten und Handbuch) in der für die MDB bereits angepaßten Version. Nach einer geringfügigen Änderung im Hello-Programm können Sie diese Neuversion des DB-Meisters übrigens auch zusätzlich auf normalen 35-Spur-Laufwerken einsetzen.

Zur Bestellung können Sie eine der im Peeker eingehefteten Bestellkarten verwenden. Stichwort:

1 x MDB10-Sonderangebot für DM 2799,-
oder
1 x MDB20-Sonderangebot für DM 3199,-

SOFTWARE SERVICE
Im Weiher 10 · 6900 Heidelberg 1

Warum wollen Sie ein teures Textverarbeitungsprogramm kaufen, wenn es ein billiges und besseres gibt?

Fast-Writer

von Harald Grumser
Programmdiskette und Handbuch
Gerätevoraussetzung: Apple IIe oder IIc (nicht II+)
DOS-3.3-Version. Auslieferung ab 1.6.86
Normalpreis DM 128,– (ISBN 3-7785-1419-9)
Sonderpreis für Peeker-Abonnenten DM 98,–

ProDOS-Version.
Normalpreis DM 128,– (ISBN 3-7785-1421-0)
Sonderpreis für Peeker-Abonnenten DM 98,–
Kombinationspreis für Bezieher der DOS-3.3-Version DM 28,–

Fast-Writer läuft auch auf neuem Apple GS!

Der Fast-Writer von Harald Grumser ist in zahlreichen Funktionen wie Scrollen, Suchen und Ersetzen mit Abstand das schnellste und damit angenehmste Textverarbeitungsprogramm für den Apple IIe oder IIc.

Flexibilität

Viele Textverarbeitungsprogramme sind geschützt und laufen deshalb nur in Verbindung mit normalen Disk-II-Laufwerken. Nicht so der Fast-Writer!

– Der Fast-Writer modifiziert weder DOS 3.3 (oder Diversi-DOS) noch ProDOS und kann deshalb mit BRUN FAST.WRITER gestartet werden. Unter Diversi-DOS ist der Fast-Writer dann in 3 Sekunden im Speicher. Vergleichen Sie einmal, wie lange es dauert, bis andere Textprogramme im Speicher sind!

– Da der DOS-3.3-Fast-Writer in den oberen 16K (= Language Card) liegt, kann man ihn vorübergehend verlassen und mit einem einfachen Befehl wieder starten. Mit anderen Worten: Der Fast-Writer ist permanent verfügbar, auch wenn Sie zwischendurch beispielsweise mit FID Dateien kopiert haben.

– Der Fast-Writer läuft mit allen externen Datenspeichern, die für DOS 3.3 oder ProDOS gedacht sind, z.B. mit dem Erphi-160-Spur-Subsystem, mit der Megaboard-MDB-Festplatte, mit RAM-Karten usw. Spezielle Anpassungen sind nicht erforderlich. Suchen Sie einmal ein Textprogramm, das mit diesen Datenspeichern auf Anhieb funktioniert!

– Der Fast-Writer kann mühelos über ein Menü für Ihre speziellen Aufgaben konfiguriert werden. Sie können z.B. per Knopfdruck die Zeilenbreite (normal 80 Zeichen) am Bildschirm einstellen, wobei ab einer Breite von weniger als 41 Zeichen automatisch auf die größere Bildschirmschrift umgestellt wird. Ferner können Sie die Größe des Arbeitsspeichers (insgesamt ca. 35 500 Zeichen) beliebig in Textspeicher und Hilfspuffer (für Löschen und Blockverschieben) aufteilen. Wenn Sie z.B. große Textblöcke im Speicher zu verschieben haben, so können Sie einen entsprechend großen Hilfspuffer von z.B. 10 000 Zeichen einrichten. Damit entfällt das zeitraubende Zwischenspeichern und Einlesen von Diskette.

Befehlsvorrat

Der Fast-Writer verfügt über eine große Zahl von Befehlen, von denen Sie in der Praxis jedoch nur wenige benötigen. Fünf Befehlsübersichten sind durch eingebaute Hilfsübersichten immer abrufbar, so daß Sie schon nach einer mehrstündigen Einarbeitung auf das Handbuch verzichten können. Eine Auswahl der wichtigsten Befehle:

– Freie Cursorbewegung in allen vier Richtungen mit eingebauter Schnell-Scroll-Routine.

– Diverse, per Knopfdruck ein- und ausschaltbare Optionen, z.B. Wortumbruch/kein Wortumbruch, Return sichtbar/unsichtbar, Kopfzeile (Statuszeile) mit Speicherbelegung, Cursorposition usw. eingeblendet/ausgeblendet, Bildschirm geteilt/ungeteilt, Tabulatorleiste sichtbar/unsichtbar, überschreibmodus (statt normalen Einfügmodus) ein/aus usw.

– Eingabe von Kontrollbuchstaben (einschließlich Ctrl-V!) möglich. Automatische Konvertierung in Groß- oder Kleinschreibung (unter Berücksichtigung der Umlaute und ß!)

– Extrem schnelles Suchen und Ersetzen von Zeichenketten (vorwärts und rückwärts).

– Makros frei definierbar und per Knopfdruck abrufbar. Makros können nicht nur stereotype Wortfolgen sein (z.B. „Sehr geehrte Herren"), sondern auch alle Befehlsfolgen, die man beim Fast-Writer sonst über die Tastatur eingeben würde. So läßt sich beispielsweise ein Text automatisch von Laufwerk 1 laden und auf Laufwerk 2 speichern.

– DOS-Kommandos wie Catalog, Delete, Rename usw. immer verfügbar (bei DOS 3.3 zusätzlich Init, bei ProDOS zusätzlich Online und Datum)

– Ausdruck auf Matrixdrucker (normal endlos), Schreibmaschine (normal mit Einzelblatt) und zu Kontrollzwekken auf Bildschirm; links- und rechtsbündig, zentriert und Blocksatz; einstellbarer linker, rechter und oberer Rand (im Text änderbar), bei Bedarf mit Kopfzeile und Paginierung usw. Der Ausdruck kann über eigene Druckertreiber umgelenkt werden, um z.B. Probleme mit Typenrädern, Steuerzeichen usw. zu beheben.

– Makros, Druckparameter, Druckertreiber und Tabulatoren können auf Diskette gespeichert werden.

Hüthig Software Service · Postfach 10 28 69 · 6900 Heidelberg 1